Novel Irregular LDPC Codes and their Application to Iterative Detection of MIMO Systems

Cornelius T. Healy

Msc. by Research

University of York

Department of Electronics

December 2010

Abstract

Low-density parity-check (LDPC) codes are among the best performing error correction codes currently known.

For higher performing irregular LDPC codes, degree distributions have been found which produce codes with optimum performance in the infinite block length case. Significant performance degradation is seen at more practical short block lengths. A significant focus in the search for practical LDPC codes is to find a construction method which minimises this reduction in performance as codes approach short lengths.

In this work, a novel irregular LDPC code is proposed which makes use of the SPA decoder at the design stage in order to make the best choice of edge placement with respect to iterative decoding performance in the presence of noise. This method, a modification of the progressive edge growth (PEG) algorithm for edge placement in parity-check matrix (PCM) construction is named the DOPEG algorithm. The DOPEG design algorithm is highly flexible in that the decoder optimisation stage may be applied to any modification or extension of the original PEG algorithm with relative ease. To illustrate this fact, the decoder optimisation step was applied to the IPEG modification to the PEG algorithm, which produces codes with comparatively excellent performance. This extension to the DOPEG is called the DOIPEG.

A spatially multiplexed coded iteratively detected and decoded multiple-input multiple-output (MIMO) system is then considered. The MIMO system to be investigated is developed through theory and a number of results are presented which illustrate its performance characteristics. The novel DOPEG code is tested for the MIMO system under consideration and a significant performance gain is achieved.

Contents

List of Tables

Acknowledgements

I would like to thank my supervisor, Dr Rodrigo de Lamare for all his help in completing my Msc. by Research. I would also like to thank my fellow postgraduate students for their help and advice along the way.

Chapter 1

Introduction

1.1 Introduction

Wireless communications has been and continues to be one of the fastest growing sectors of technology. With the ever increasing proliferation of smartphones, effectively handheld computers capable of fully utilising internet connectivity with integrated mobile phone functionality, and wireless local and personal area networks, the demand for increasingly high data rate, the need for reliable transmission is constantly growing. Error control coding is an integral element in any practical communication system. Low-density parity-check (LDPC) codes are among the best performing codes currently known.

LDPC codes are a class of capacity-approaching codes first introduced by Gallager [1] which were largely ignored for decades due to the computational complexity of their implementation. A notable exception being Tanner's paper [2] in which he introduced the useful graphical representation of the parity-check matrix which bear his name, Tanner graphs. Luby et al. [3] extended the concept of LDPC codes to the irregular case, showing that by allowing varied row and column weights an improvement in performance may be seen. Richardson et al. [4], with Density Evolution (DE), provided a method to derive optimal degree distributions for codes of infinite length,

subject to certain conditions.

Given the analytically optimal degree distributions for the ideal infinite block length case considered using DE, considerable effort has been invested in finding methods to implement LDPC codes at more practical block lengths without sacrificing the excellent performance characteristics of longer codes. With finite length codes, particularly at short to medium lengths, the assumption in DE that the decoding neighbourhood of a given variable node is tree-like [4] no longer holds. This means that we can no longer assume full independence of messages passed in sum-product/belief propagation decoding. This manifests as cycles in the parity-check matrix/Tanner graph of the code. A significant focus in the search for practical finite-length codes is the mitigation of the effects of these cycles which break down the independence assumption.

Of the cycles which exist in finite length codes, it has been noted that the length of the shortest cycle of the code (the girth of the code) has a significant effect on its performance. In fact, for iteratively decoded LDPC codes, the number of independent iterations of the message passing algorithm used is proportional to the girth of the code [1].

While algorithms exist which perform girth conditioning on constrained randomly generated LDPC codes, among those codes capable of best performance at practical lengths are codes designed by the Progressive Edge Growth (PEG) algorithm [5], along with modifications to this algorithm [6][7]. The PEG algorithm is a greedy edge placement construction method for the parity-check matrix of an LDPC code which places edges in the Tanner graph of the code such that when a cycle is created, that cycle is of the maximum possible length under the current graph settings. This algorithm produces LDPC codes with relatively large girth and with particularly large local girth in the lower weight variable node subgraph of the parity-check matrix, leading to improved performance.

Another approach to constructing good finite length LDPC codes is, rather than increasing the girth of the code, to increase the connectivity between the cycles present in the graph of the code. This higher connectivity allows greater transmission of ex-

trinsic messages in iterative decoding with the sum-product algorithm, thus mitigating the effect of the cycles involved on the performance of the code. This concept led to the definition of the Extrinsic Message Degree (EMD) of a node in the graph of a code, and subsequently to the easily calculable Approximate Cycle EMD (ACE) metric of connectivity of a cycle [8].

The promising performance of both the PEG algorithm and the design method presented by Tian et al. [8] naturally led to the combination of both concepts, leading to the improved PEG (IPEG) algorithm [6] among others [7].

In this thesis, a further improvement to the PEG algorithm based construction methods is proposed. This modification is based on the application of the iterative (SPA) decoder at a key stage before edge placement in the PEG algorithm in order to identify which edge, from a number of candidates provided by the algorithm, will produce the best performance under the current graph setting. This optimised selection of edges for placement leads to significant improvement of performance over existing methods in the short to medium length.

The code generated by this novel construction method is then applied to a multiple-input multiple-output (MIMO) system. The MIMO system, with multiple transmit and receive antennas and introduced in the work by Foschini [9] and Telatar [10], can provide significant increase in capactiy for a given wireless channel. This increase in capacity results from the exploitation of spatial multiplexing and spatial diversity at the transmit and receive antennas.

1.2 Goals

The objectives of this thesis are as follows:

- to provide an overview of regular and irregular LDPC codes in general, detailing their performance characteristics, along with a more detailed decription of some high performance LDPC codes in particular, including codes constructed by the Progressive Edge Growth (PEG) and the improved PEG (IPEG) algorithms.

- to present a novel construction method for irregular LDPC codes based on a modification of the PEG algorithm, showing a significant improvement in performance over existing methods.

- to demonstrate the performance gain achieved by the novel code presented herein in the case of a MIMO system and its use in the design of iterative detection and decoding algorithms for interference mitigation.

1.3 Contributions

The primary contribution of this thesis is the proposed design method for irregular LDPC codes. Applying this design method, extra effort at the design stage provides significant improvement in performance at no cost of increased complexity during operation.

The second contribution presented in this thesis is an analysis of the above novel code in operation in an iteratively detected and decoded spatially multiplexed MIMO system.

1.4 Thesis Layout

The remainder of this thesis is organised as follows. In Chapter 2 LDPC codes are briefly overviewed, with a general introduction to the LDPC coding system followed by a review of the literature relevant to the discussion of the following chapters.

In Chapter 3 the novel construction method which is the main focus of this thesis is described and results are presented and analysed.

In Chapter 4 a MIMO communication system is briefly described and the code developed in the previous chapter is applied to the case of iterative detection and decoding of a MIMO system.

Chapter 5 provides the conclusions of this thesis.

Chapter 2

Review of LDPC Codes

In this chapter, we review LDPC codes. In Section 2.2 the LDPC coding system is introduced. The parity-check and generator matrices of the code are defined along with the notation used. The graphical interpretation of the parity-check matrix, the Tanner graph, is then presented. The concept of degree distributions for describing irregular LDPC codes is detailed before a short review of the iterative approach to decoding LDPC codes.

In Section 2.3 a review of the literature of LDPC codes is carried out. A number of explicit construction methods for the parity-check matrix are reviewed, along with a number of concepts which form the basis of the original work presented in Chapter 3.

2.1 Linear Block Codes

In the binary field, an (n, k) block code is a set of 2^k length-*n* vectors, called codewords, uniquely corresponding to the 2^k possible permutations of a length-*k* message vector.

The block code is said to be linear if the modulo-2 sum of any two codewords produces a third codeword. Associated with every linear block code are the matrices

G, the generator matrix, and **H**, the parity-check matrix, such that the codeword **c** is related to the message vector **m** by the expression

$$\mathbf{c} = \mathbf{mG}, \tag{2.1}$$

The generator matrix and parity-check matrix satisfy

$$\mathbf{GH}' = 0, \tag{2.2}$$

and

$$\mathbf{cH}' = 0 \tag{2.3}$$

An encoder is systematic if the codeword has the form

$$\mathbf{c} = [\mathbf{p}\ \mathbf{m}] \tag{2.4}$$

where **m** is the message vector as indicated above and **p** is the vector of parity bits. The systematic generator matrix may then be written in the form

$$\mathbf{G} = [\mathbf{P}'\ \mathbf{I_k}] \tag{2.5}$$

where $\mathbf{I_k}$ is the identity matrix of size *k* and **P** is an (*n*-*k*)-by-*k* matrix. Then a corresponding parity-check matrix may be determined as

$$\mathbf{H} = [\mathbf{I_{n-k}}\ \mathbf{P}], \tag{2.6}$$

2.2 LDPC Coding System

LDPC codes are linear block codes fully characterised by the (*n*-*k*)-by-*n* sparse parity-check matrix, **H**.

Figure 2.1: LDPC Coding System

As shown in Fig. 2.1 above, the codeword **c** is derived from the message vector **m** as in Eqn. 2.1 for conventional block codes. The LDPC code may then be decoded by the SPA algorithm which makes use of the structure of the parity-check matrix.

If the parity-check matrix is obtained in non-systematic form, **B**, the systematic generator and parity-check matrices may be found as follows:

Gaussian elimination (GE) with column pivoting is used to determine the (*n*-*k*)-by-(*n*-*k*) matrix $\mathbf{A_p^{-1}}$ such that

$$\mathbf{H} = \mathbf{A_p^{-1}A} = [\mathbf{I_{n-k}}\ \mathbf{P}] \tag{2.7}$$

where **A** is derived from **B** simply by the rearrangement of columns required by GE. **G** is constructed as per Eqn. 2.5 and so

$$\mathbf{HG'} = 0 \Rightarrow \mathbf{A_pHG'} = 0 \Rightarrow \mathbf{AG'} = 0 \tag{2.8}$$

Now we have the parity-check matrix **A** in systematic form, and as will be become clear from the introduction of the graphical representation of LDPC codes in the following section, along with the discussion of Section 2.3.2, the required rearrangement of columns will not affect code performance. It should be noted that **H** in the form of Eqn. 2.7 is not necessarily sparse and is not in a form conducive to decoding by the iterative message passing algorithm to be introduced in Section 2.2.3.

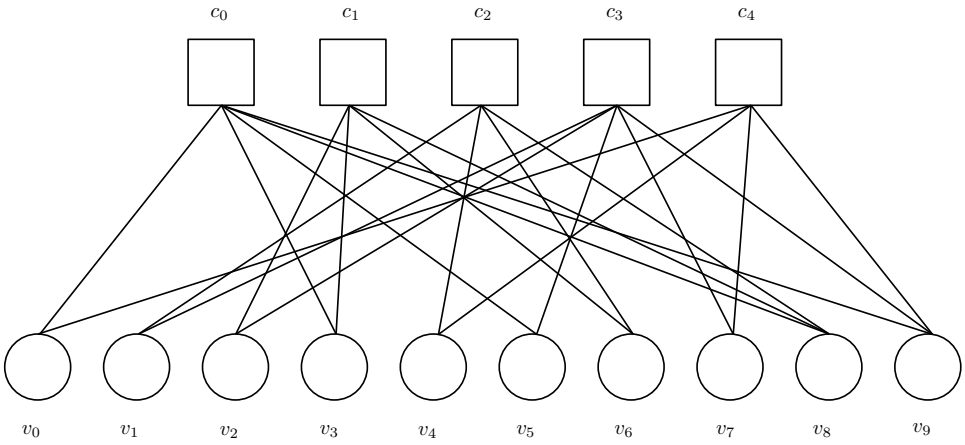
2.2.1 Graphical Representation

A graphical interpretation based on a bipartite graph, now referred to as a Tanner graph was provided by Tanner [2]. The graph consists of two types of nodes, variable nodes

and check nodes, connected by edges. There are n variable nodes $\{v_i; i = 1, \ldots, n\}$ and $m = n - k$ check nodes $\{c_j; j = 1, \ldots, m\}$. An edge connects variable node i to check node j if there is a 1 in the position (j, i) of the parity-check matrix, **H**. An example parity-check matrix (not sparse) and its corresponding Tanner graph are shown in Fig. 2.2 below.

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

(a)



(b)

Figure 2.2: (a) Parity-check Matrix
(b) Corresponding Tanner Graph

2.2.2 Irregular Degree Distributions

LDPC codes, as defined by Gallager [1], have parity-check matrices with fixed column and row weights, d_v and d_c, respectively. Such an arrangement is now referred to as a regular LDPC code. Subsequently Luby et al. [3] introduced irregular LDPC codes with row and column weights which varied according to their degree distributions, defined as:

$$\lambda(x) = \sum_{i=0}^{d_v} \lambda_i x^{i-1} \tag{2.9}$$

$$\rho(x) = \sum_{j=0}^{d_c} \rho_j x^{j-1} \tag{2.10}$$

where:

- λ_i is the fraction of all edges connected to degree-i variable nodes,

  $$0 \leq \lambda_i \leq 1, \quad i \geq 0, \quad \sum_{i=0}^{d_v} \lambda_i = 1$$

  and here d_v is the maximum variable node degree.

- ρ_j is the fraction of all edges connected to degree-j check nodes,

  $$0 \leq \rho_j \leq 1, \quad j \geq 0, \quad \sum_{j=0}^{d_c} \rho_j = 1$$

  and d_c is the maximum check node degree.

In the previous example in Fig. 2.2, the given parity-check matrix has

$$\lambda(x) = 0.8x + 0.2x^2 \tag{2.11}$$

$$\rho(x) = 0.6x^3 + 0.4x^4 \tag{2.12}$$

2.2.3 Iterative Decoding Procedure

LDPC codes are decoded using iterative decoding techniques in which the two different types of nodes of the Tanner graph effectively behave as two separate serially concatenated decoders. At each iteration of the overall decoder, each constituent decoder sends extrinsic information to the other constituent decoder, calculated using the information received from the other constituent decoder in the previous iteration as intrinsic input information. By means of this message passing strategy and utilising the dependencies between codeword bits introduced by the encoding procedure, errors introduced by channel noise may be corrected as the decoder converges.

In practice, the message passing algorithm propagates log-likelihood ratios (LLRs) in order to avoid computationally costly multiplications and to avoid numerical instability which may arise when computing iteratively with probabilities. The log-domain sum-product algorithm (SPA) is used. The usefulness of the Tanner graph representation of LDPC codes now presents itself, as messages passed during each iteration of the decoder may be viewed as being sent from a variable node to a check node and from a check node to a variable node along the edges of the Tanner graph.

At one half iteration, the LLR sent from a variable node (VN) v_i, $\{i = 1, ..., n\}$, to a check node (CN) c_j, $\{j = 1, ...d_{v_i}\}$, connected to it is:

$$L_{i \to j} = L_{ch,i} + \sum_{j' \neq j} L_{j' \to i} \tag{2.13}$$

where $L_{j' \to i}$ is the LLR received from CN j' to VN i in the previous iteration and, for channel output y_i corresponding to transmitted coded bit $x_i \in \{\pm 1\}$,

$$L_{ch,i} = \log \left(\frac{p(x_i = +1|y_i)}{p(x_i = -1|y_i)} \right) \tag{2.14}$$

In the other half iteration, the LLR sent from CN j, j=1,...,m, to VN i, i=1,...,d_{c_j}, connected to it is:

$$L_{j\to i} = 2\tanh^{-1}(\prod_{i'\neq i}\tanh(L_{i'\to j}/2)) \tag{2.15}$$

where $L_{i'\to j}$ is the LLR received from VN i' to CN j in the previous iteration.

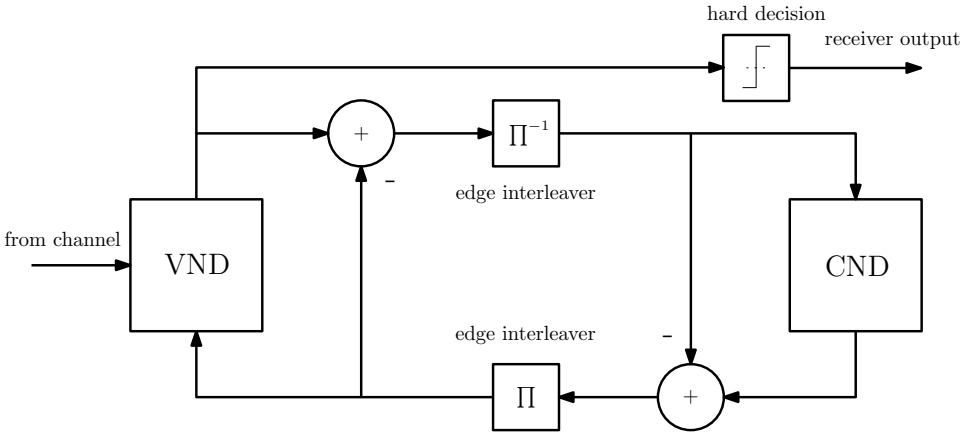


Figure 2.3: Block Diagram of the Message Passing LDPC Decoding

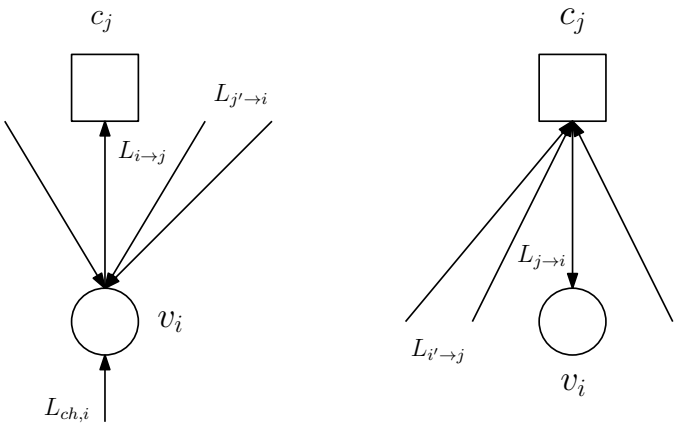


Figure 2.4: Diagram of the Messages Passed at Each Half-Iteration in SPA Decoding

In Fig. 2.3 the block diagram of the overall structure of the iterative LDPC decoder is shown. As in the description above, decoding is viewed as a process of exchanging

iteratively updated messages between two simple decoders, the variable node decoder (VND) and the check node decoder (CND). The VND carries out the operation of Eqn. 2.13 for each node. At the first iteration, assuming encoded bits are equally likely to be “+1” and “-1”, the second term of Eqn. 2.13, the *a priori* LLRs, are 0 and so the operation is carried out using information from the channel only. In subsequent iterations the *a priori* information is utilised in computing the messages to be sent as indicated. To ensure the VND and CND operations remain independent for as long as possible, the *a priori* information is removed from the *a posteriori* LLRs before sending them to the CND. Likewise after the CND operation intrinsic information is removed, extrinsic information only is sent on for use in the next iteration. The edge interleavers represent the interconnections of the Tanner graph.

Fig. 2.4 shows the messages passed at each variable node and each check node in the Tanner graph at each half iteration.

2.3 Literature Review

In this section, the literature concerning construction of the LDPC matrix is reviewed. First the regular construction methods of Gallager and Mackay are detailed. Following this, a discussion of the challenges of constructing an irregular LDPC matrix which provides improved performance is provided. A number of construction methods for the regular and irregular cases are then reviewed.

2.3.1 Gallager and MacKay Construction Methods

Gallager Codes

In his original paper [1] Gallager proposed a construction method for regular (n, j, k) codes, where n is block length, j is the regular column weight and k is the regular row weight. The construction method is based on random column permutation of a base matrix $\mathbf{H_1}$ with $\frac{n}{k}$ rows and column weight 1 which has the following simple structure. For $i = 1, 2, \cdots \frac{n}{k}$ the *i*-th row of $\mathbf{H_1}$ has all its 1's in columns (*i*-1)*k*+1 to ik. That is the first row has 1's in positions $1 \cdots k$, the second row has 1's in positions $k+1 \cdots 2k$ and so on. The submatrices $\mathbf{H_2}$ to $\mathbf{H_j}$ are constructed simply by column permutations of $\mathbf{H_1}$. The parity-check matrix $\mathbf{H}$ is then constructed as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H_1} \\ \mathbf{H_2} \\ \vdots \\ \mathbf{H_j} \end{bmatrix}$$

MacKay Codes

MacKay provided and analysed the performance of a number of constrained random construction methods of increasing complexity with increasing constraints [11]. The simplest construction method presented constrains block length, column weight and

row weight as in Gallager codes and has the added condition that no two columns have an overlap greater than 1. This is equivalent to the condition that there be no cycles of length 4 in the graph of the code. MacKay extended this construction method to exclude longer cycles and also to the case of irregular variable node degree distributions.

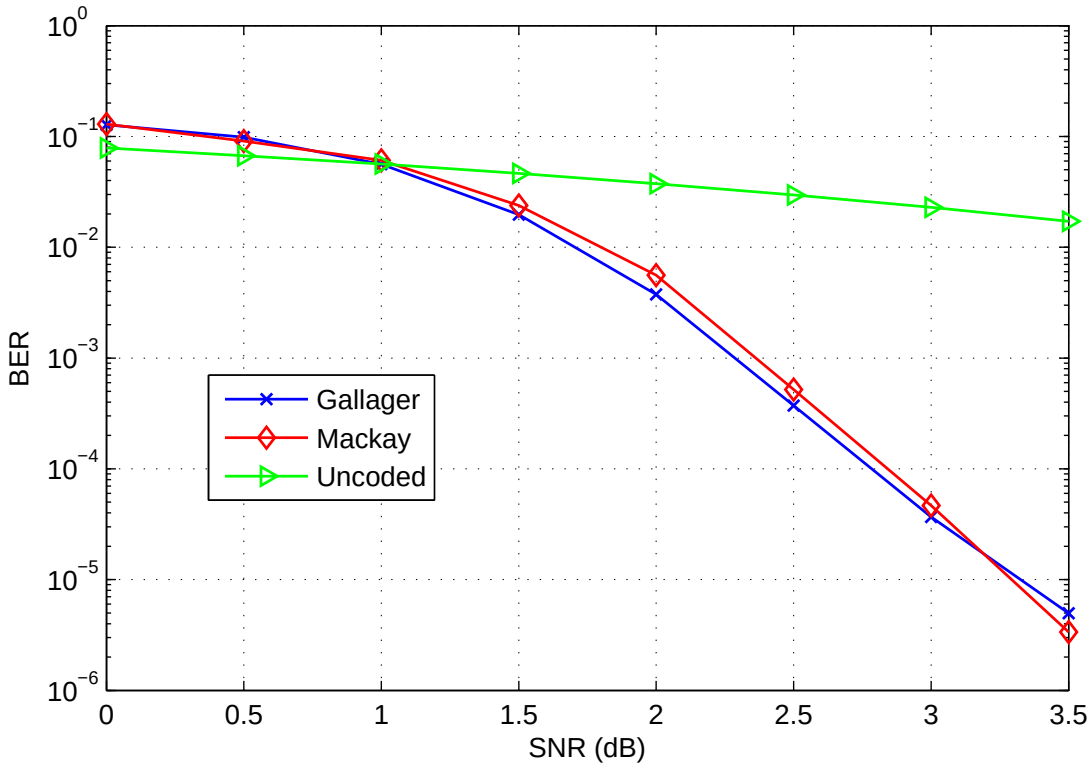


Figure 2.5: Comparison of Performance of Gallager and Mackay codes for length $n = 500$

In Fig. 2.5 above, the performance of the codes constructed by the methods described in Section 2.3.1 is shown and compared with the uncoded case in the additive white Gaussian noise (AWGN) channel. BPSK modulation was used along with SPA decoding in the case of the coded transmissions. The decoder was operated to a maximum of 50 iterations and 100 block errors were gathered per SNR point.

2.3.2 Density Evolution

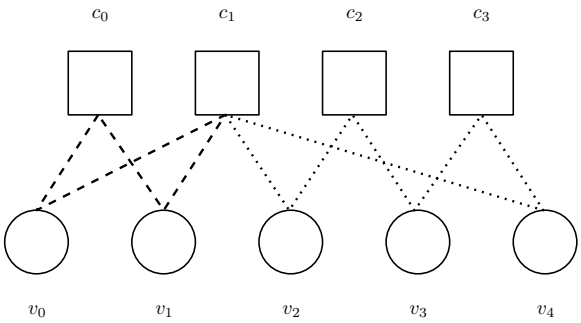
In Density Evolution (DE), Richardson and Urbanke [4] provided a method to compute the optimal degree distributions for codes of infinite length under certain assumptions, by means of analysing the evolution of error probability in the message passing decoder as it progresses through iterations. For a given pair of degree distributions the threshold, i.e., the worst channel paramater such that the probability of error converges to zero as the number of iterations tends to infinity, is computed. Using search methods, pairs of degree distributions (λ, ρ) were identified which maximised the threshold value. This method has proved to be a valuable tool and for codes of very large block length, the performance exhibited is very impressive.

However, for codes of short to medium length, which are more practical in systems where latency is an issue, the assumption of DE that the decoding neighbourhood of a given VN is tree-like - valid for the case of codes of infinite length - no longer holds. In this case cycles are present in the Tanner graph of the code. A cycle is a path through the graph which originates and ends at the same node without traversing any single edge twice. This is illustrated in Fig. 2.6 for both the parity-check matrix and the Tanner graph of an example code.

Cycles in the graph of an LDPC code degrade performance - with cycles present in the graph, after a number of iterations of the decoder, the messages passed will no longer be fully independent. The length of the shortest cycle of an LDPC code is called the girth of the code. Length 4 cycles, as seen in Fig. 2.6 (a) occurring between variable nodes v_0 and v_1 and check nodes c_0 and c_1, the dashed set of edges in the Tanner graph in Fig. 2.6 (b), are the shortest cycles possible and are also the most damaging in terms of performance. It is usual for all codes designed to perform some girth conditioning to remove cycles of length 4. Also shown, occurring between VNs v_2 to v_4 and CNs c_1 to c_3 is a cycle of length 6, highlighted as the dotted set of lines in the Tanner graph of Fig. 2.6 (b).

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

(a)



(b)

Figure 2.6: (a) Parity-check matrix and
(b) Tanner graph illustrating cycles of length 4 and length 6

In addition to removal of length-4 cycles, another common deviation from the prescribed optimal degree distribution pairs of DE when designing practical LDPC codes is to limit the number of variable nodes of weight 2 to less than *m*, the number of check nodes of the code. This constraint limits (and for the PEG algorithm described later removes entirely) the possibility that cycles exist which are made up only of weight-2 VNs. Without this constraint, cycles made up of only weight-2 nodes will exist in the graph [8]. This type of cycle is particularly damaging to performance as they have no connection to the graph outside of the cycle, and so receive no extrinsic information from the rest of the graph, leading to poor performance in the error floor region.

2.3.3 Approximate Cycle EMD

The concept of connectivity of cycles introduced above forms the central idea of the Approximate Cycle EMD (ACE) metric defined by Tian et al. [8][12]. They state that the connectivity of the cycles and not simply the length of the cycles present in the graph of a code determine the performance of the code. The worst-case scenario in terms of connectivity of a cycle is a stopping set. This is defined as a set of VNs for which every CN connected to a VN in the set is connected to the set at least twice. In practice this means either the cycle is either made up of only weight-2 VNs or is comprised of a number of cycles connected together. Performance of LDPC codes under iterative message-passing decoding is directly related to how the constituent cycles of its graph connect to form stopping sets. This has been shown explicitly for the binary erasure channel (BEC) [13]. The error performance of LDPC codes over the BEC may be completely determined given the stopping sets of the Tanner graph of the code and the erasure probability, ϵ. Stopping sets are further discussed in [14], where is shown that their influence on performance translates to the additive white Gaussian noise (AWGN) channel, with code bit LLRs with poor reliability considered rather than erasures.

Tian et al. define the Extrinsic Message Degree (EMD) of a VN set as the number of CNs singly connected to that set. For a cycle in which no two VNs share CNs outside the cycle (ie. there exists no sub-cycle) the EMD of the cycle is $\sum_i (d_i - 2)$. For convenience of calculation, this case in which the cycle in question is assumed to have no sub-cycle is considered and the metric is labeled the Approximate Cycle EMD (ACE).

The design method proposed in [8] was as follows: column by column generation of the parity-check matrix, with each column generation followed by computation of the ACE metric and a check to see that it meets or exceeds a prescribed minimum. If it does, the column is retained and if not the column is discarded and random generation carried out again. This results in codes which outperformed codes generated by

conventional constrained random generation followed by girth conditioning as shown in [8]. However, codes generated by this construction method are themselves outperformed by the Progressive Edge Growth algorithm to be described later, as shown in [15].

As a result, the construction method of [8] is not considered further. However, the ACE metric proves to be a useful measure of connectivity of cycles in a graph and the ideas presented prove useful in understanding the effect of cycles in the graph on the performance of an LDPC code.

2.3.4 Repeat Accumulate Class of Codes

The iterative decoding procedures for LDPC codes described in Section 2.2.3 , benefitting from the sparseness of the parity-check matrix, provide high performance in acceptable computational complexity, which is essentially linear with the block length n. A considerable drawback of the LDPC coding system is the fact that the encoding described by equation 2.3 involves the generator matrix, $\mathbf{G}$, which is in general not sparse. As a result encoding complexity is high.

A number of approaches have been presented which tackle this issue by imposing structure on the parity-check matrix, which then may be exploited for faster encoding. Examples of this structure include upper/lower triangular forms [16][5] and cyclic and quasi-cyclic codes [17][18]. These approaches generally involve a tradeoff between performance and encoding complexity, with decoding complexity also increasing in the finite geometry based quasi-cyclic case [19].

A class of codes was presented in [20] and expanded upon [21], repeat accumulate (RA) codes and their irregular counterparts known as irregular repeat accumulate (IRA) codes. These codes may be viewed as both serial turbo codes and LDPC codes. That is, they may be viewed in terms of an LDPC matrix and decoded accordingly by means of the efficient iterative SPA decoder, and they may also be viewed in terms of a pair of codes, outer repeat and inner accumulate codes separated by an interleaver,

and thus can be encoded efficiently as such, with complexity increasing linearly with the block length n.

The block diagram for the encoder of the RA and IRA codes is shown below, where for the regular case, the matrix operation of **A** is replaced by a simple repeat code. The interleaver, $\prod$, describes the connections in the Tanner graph of the code when SPA decoding is employed. The final element of Fig. 2.7 is a simple rate-1 convolutional code, called an accumulator. The block, T, is simply a delay element. The dashed line indicates the systematic version of the IRA code, in which case the matrix **A** would have dimensions *k*-by-(*n*-*k*) and the output vector of the accumulator **w** would be *1*-by-(*n*-*k*).

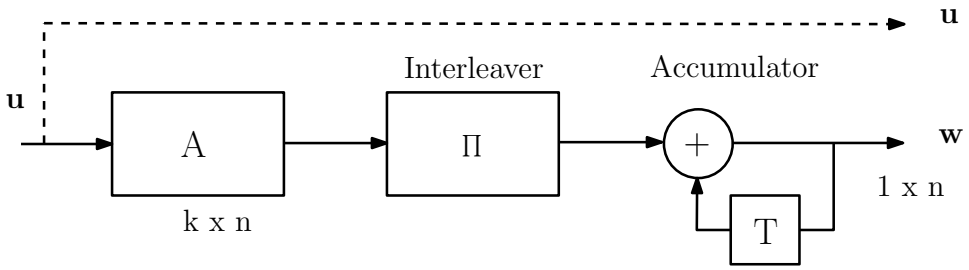


Figure 2.7: Encoder for the RA/IRA codes

Extended IRA Codes

In [22], Yang et al. explicitly show the low-density parity-check matrix interpretation of the IRA code class. A discussion of the vulnerability of weight-2 variable nodes is developed into the definition of a new subclass of codes, extended IRA or eIRA codes. These codes are capable of efficient encoding as a result of their IRA basis and also capable of excellent performance at high rates. A plot is provided showing the evolution of expected LLR magnitudes of the code produced by the construction method proposed in the paper. This plot provides a means of comparing the performance of variable nodes of different weights over a number of iterations. As expected the low weight variable nodes converge more slowly and to a lower magnitude as they receive less information upon which to operate. As is discussed, low weight VNs are required

in order to allow for lower weight check nodes as is required by the optimal irregular degree distributions of [4]. Low weight CNs are less likely to result in a check operation failure. x The dual diagonal matrix, also shown in [8] to be cycle free is utilised with the inclusion of a single weight-1 column. This makes up part of an LDPC matrix. The other part is specified to be free of weight-2 columns, low density and irregular such that the overall distribution of the parity-check matrix is near-optimal. Then the parity-check matrix has the form

$$\mathbf{H} = [\mathbf{H_1}\ \mathbf{H_2}]\,, \tag{2.16}$$

where $\mathbf{H_2}$ is the matrix of the above form, dual diagonal with appended weight-1 column. Now the generator matrix is shown to have the form

$$\mathbf{G} = [\mathbf{I}\ \mathbf{P}] = \left[\mathbf{I}\ \mathbf{H_1^T H_2^{-T}}\right], \tag{2.17}$$

where $\mathbf{H_2^{-T}}$ is in upper triangular form and represents the accumulator of Fig. 2.7. In fact LDPC encoding with **G** above may be carried precisely as in Fig. 2.7 except that here the matrix **A** has the form $\mathbf{H_1^T}\Pi^{-}1$. This is a low density matrix and so encoding has low complexity. In fact this code possesses the excellent encoding properties of turbo codes and the decoding properties of LDPC codes.

Accumulate Repeat Accumulate (ARA) Codes

In [23], a class of codes is developed which is presented as an enhanced extension of the RA class of codes, where precoding with another accumulator is carried out. This serves to improve the input-output extrinsic SNR behaviour of the code in the high extrinsic SNR region. These ARA codes are also presented in protograph structure, a protograph being a graph with a relatively small number of nodes which defines the code and from which the code is produced by a copy and permute operation. The protograph approach to ARA codes is further developed in [24] and in [25] for low rate codes.

2.3.5 Progressive Edge Growth Algorithm

.

The PEG algorithm is a construction method for LDPC codes which, given a variable node degree sequence, block length and code rate produces codes which are among the best performing codes currently known. For irregular LDPC codes, the degree sequences which are found to provide the best performance are those derived from the density evolution optimisation procedure. The PEG algorithm is a highly flexible in that it may be used to generate codes of any block length, rate and for any given variable node degree distribution.

From a graphical viewpoint, the algorithm progresses on an edge-by-edge basis, or equivalently, the algorithm places the "1"s in the parity-check matrix of the code, one entry at a time. Edge placements are made such that when a cycle is created, that cycle is of maximum possible length under the current graph settings. This approach ensures large overall girth for the code. Additionally, it ensures particularly large girth in the left-hand sub-graph of the code. According to the algorithm, the input degree sequence is arranged in non-decreasing order and the edge placements are made in progression from left to right in the graph/parity-check matrix. As such the low-weight variable nodes, in the left-hand sub-graph, are imparted with particularly large girth. As previously discussed in Section 2.3.3, short cycles among low-weight variable nodes are very damaging to performance. Combined with the large overall girth properties and minimum distance bound greater than that of randomly generated codes, this is the source of the excellent performance of PEG generated codes.

Definitions and Notations

- n - block length of the parity-check matrix to be generated.
- D_s - Variable node degree sequence, the weight of the columns of the parity-check matrix to be generated, in non-decreasing order. This is related to $\lambda(x)$ defined in Section 2.2.2 by

$$D_s = [d_{min}...d_a...d_{max}], \tag{2.18}$$

where

$$d_a = n\frac{\frac{\lambda_a}{a}}{\int_0^1 \lambda(x)dx} \tag{2.19}$$

- v_j - variable node j, $j = 1, ..., n$
- c_i - check node i, $i = 1, ..., m$
- $N_{v_j}^l$ - the neighbourhood of node v_j to depth *l*. This is defined as the set of check nodes which may be reached by a subtree starting from node v_j and expanding for *l* levels, where each level consists of variables nodes at equal distance from v_j and all the check nodes connected to them. This is illustrated in Fig. 2.8.
- $\overline{N_{v_j}^l}$ - the set of all check nodes excluding those in the neighbourhood of node v_j to depth *l*, $N_{v_j}^l$.

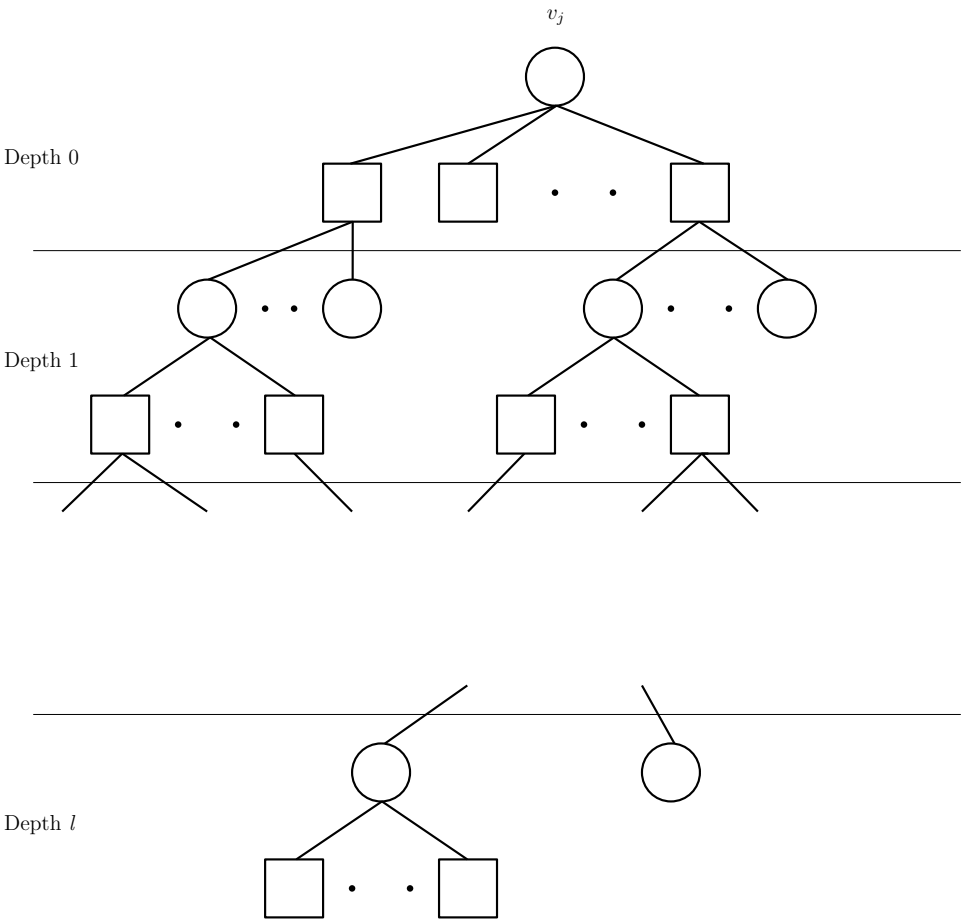


Figure 2.8: Neighbourhood of v_j to depth *l* [5]

PEG Construction

The graph is initialised with n variable nodes, m check nodes and no edges. Placements are made edge by edge, and progressing through columns from left to right, according to the following procedure.

For the first edge emanating from a variable node, the edge is placed connecting the variable node to the lowest weight check node under the current graph setting. If more than one such lowest weight check node exists, the candidate is chosen at random from the set.

For each subsequent edge to be placed, a subtree (see Fig. 2.8) is expanded from the variable node in question up to the depth that either:

(a) the tree expands further but fails to include any extra check nodes.

(b) the next step in the tree expansion will include all check nodes in the tree.

In the case of (a), the set of nodes not currently in the tree cannot be reached from the current variable node, and as such when a placement is made connecting the current variable node to one of the check nodes in the set $\overline{N_{v_j}^l}$, those check nodes not currently in the tree, no cycle is created. For the PEG algorithm a further restriction on choice of placement is made. The set of nodes in $\overline{N_{v_j}^l}$ with minimum weight is referred to as the set of candidate check nodes. The edge is placed connecting the current variable node of interest to a check node in the candidate check node set. If there is more than one node in this set, a candidate is chosen at random.

In the case of (b), all nodes in the graph can be reached from the variable node of interest, and when an edge is placed, a cycle will be created. However, as the candidate set is taken as the minimum weight check nodes of the CNs not currently in the tree at the point at which one more level-expansion will result in the tree including all CNs, the cycle created will be of maximum length possible, which is length 2*(*l*+2). As above if there is more than one check node in the set of candidates the choice is made at random from the set.

The pseudocode for the PEG algorithm is presented in Table 2.1.

A Note on CN Degree Distribution

The optimal degree distributions produced by density evolution, as introduced in Section 2.2.1 consist of the pairs $\lambda(x)$, $\rho(x)$, defining the variable node and check node degree distributions respectively. However, as stated above, of these pairs, the PEG algorithm takes as an input only the variable node degree sequence derived from the VN degree distribution by expressions (2.18) and (2.19). The check node degree distribution of the code generated by the PEG algorithm, by virtue of the "minimum weight check node" condition applied at every choice among candidates, is as uniform as possible. This tends to result in a CN degree distribution of, or very close to the form

$$\rho(x) = sx^t + (s-1)x^{t+1} \tag{2.20}$$

$$\text{for some } t \geq 2 \text{ and } 0 \leq s \leq 1$$

As stated by [5], evidence exists to suggest that this concentrated degree sequence for check nodes is optimum.

Table 2.1: Pseudocode for the PEG Algorithm

For $j = 1$ to n

For $k = 1$ to $D_s(j)$

If $k == 0$

place edge between current VN v_j and CN c_i such that $c_i \in$ (the set of CNs with minimum weight under the current graph setting).

Else

expand tree to depth *l* under current setting s.t. the cardinality of $N_{v_j}^l$ stops increasing but is less than m **or**

$\overline{N_{v_j}^l} \neq \emptyset$ but $\overline{N_{v_j}^{l+1}} = \emptyset$

Then place edge between current VN v_j and CN c_i s.t. $c_i \in \overline{N_{v_j}^l}$ with lowest CN degree. If a number of CN candidates meet this requirement, choose one at random.

End If

End For

End For

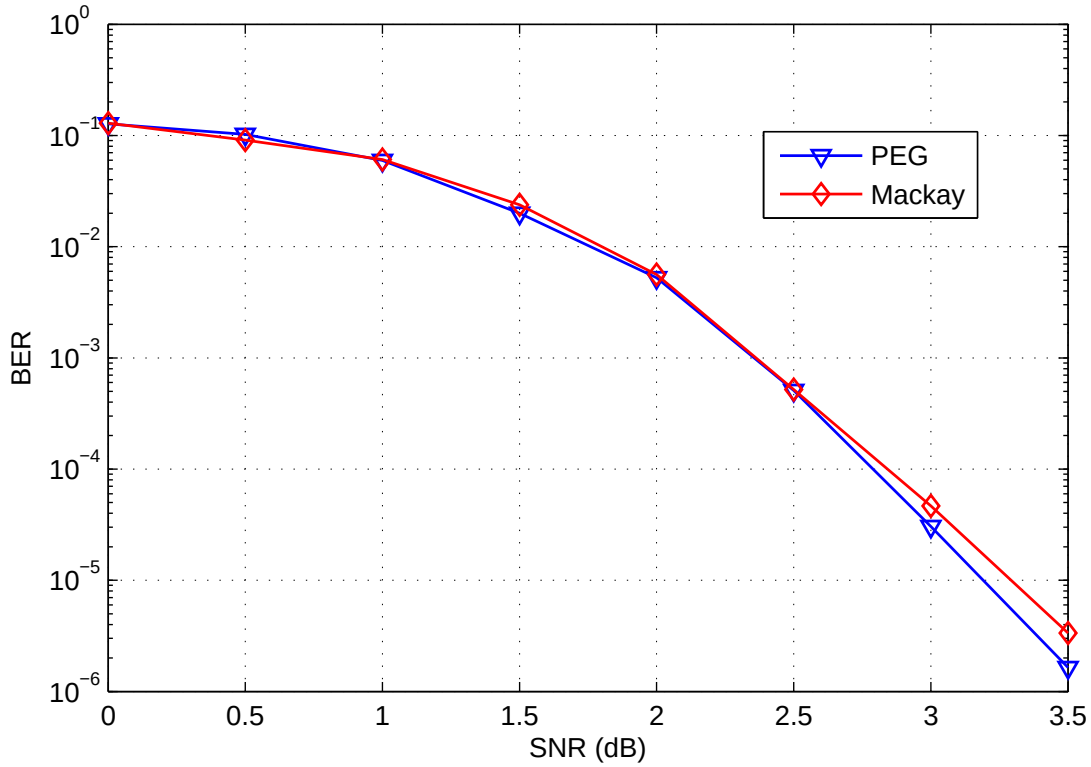


Figure 2.9: Comparison of performance of regular PEG and MacKay generated codes of length $n = 500$

In Fig. 2.9 above, a comparison of the performance of length $n = 500$ rate $\frac{1}{2}$ (3,6)-regular codes constructed by both the Gallager method described in Section 2.3.1 and the PEG algorithm is presented. BPSK modulation was used and the codewords were decoded by the SPA algorithm under the presence of additive white Gaussian noise. The regular PEG generated code outperforms the Gallager code. However, as will be seen in the following Fig. 2.10, the PEG code with optimal irregular degree distribution outperforms both the (3,6) regular PEG and Gallager codes.

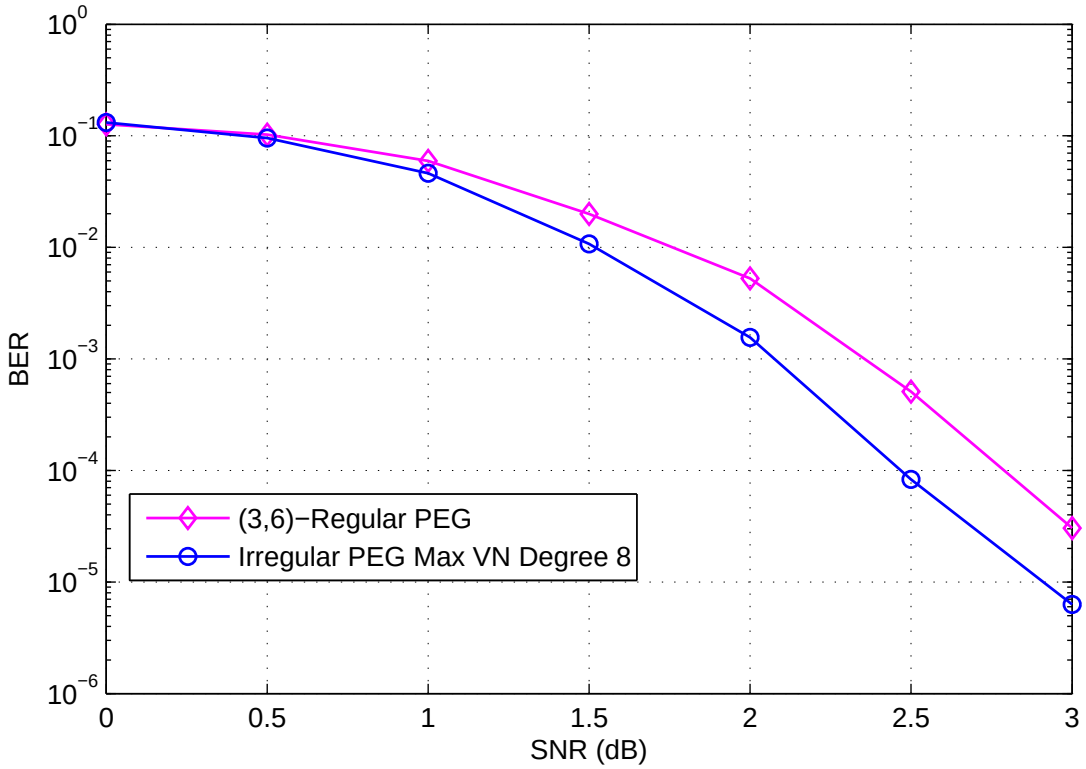


Figure 2.10: Comparison of performance of regular and irregular PEG generated codes of length $n = 500$

2.3.6 Improved Progressive Edge Growth Algorithm

The Improved Progressive Edge Growth (IPEG) algorithm [6] is an extension to the PEG algorithm which incorporates the ACE metric, introduced by Tian et al [8] and discussed in Section 2.3.3, in the edge selection procedure of the PEG algorithm. By utilising the ACE metric to aid in selection of the edge to be placed at each step of the algorithm, the candidate chosen has the greatest connectivity to the rest of the graph.

As stated in [8], greater connectivity among subgraphs of the graph of an irregular LDPC code leads to lower error floors in the bit error rate (BER) curve of the code. This improvement is clearly demonstrated in the comparison of BER plots for codes generated by the PEG and IPEG construction methods, with identical variable node degree sequences and rate, as shown in Fig. 2.11. The system is as described for previous results, namely BPSK modulation, AWGN channel and SPA decoding in the receiver.

ACE Metric Calculation

The ACE metric measures connectivity of a cycle to the rest of the graph. In the case of the IPEG algorithm, the connectivity of each cycle which would be created by placing each candidate check node is compared, and the cycle which has greatest connectivity is chosen. As with the original PEG algorithm which maximises the length of the cycle which must be created under the current graph setting, the IPEG algorithm uses the ACE metric to choose the candidate which will create the least damaging cycle possible. Connectivity of the cycle is measured by counting the number of edges by which the cycle is connected to the rest of the graph through its variable nodes [6]. Each variable node is represented by a column of the parity-check matrix, and so this count is carried out by summing the weights of the columns which represent the VNs involved in the cycle of interest. That is

$$\sum_{v_p}(w_{v_p}-2), \tag{2.21}$$

where w_{v_p} is the weight of the column v_p and the summation is taken over all the variable nodes involved in the cycle. This is precisely the approximate cycle EMD (ACE) metric of [8], where EMD stands for extrinsic message degree. The ACE metric is approximate because, for ease of calculation, the case of a single check node being shared by more than one variable node in the cycle is neglected.

In the IPEG algorithm, the ACE metric calculation is carried out for each candidate check node in the event that the candidate will create a cycle when the edge is placed. That is for the case when the subtree expansion has been terminated by the condition

$$\overline{N_{v_j}^{l}} \neq \emptyset \quad \text{but} \quad \overline{N_{v_j}^{l+1}} = \emptyset \tag{2.22}$$

being met. Then the candidate with the highest ACE metric for its associated cycle is chosen. In the case that more than one cycle has this maximum ACE metric, a candidate is chosen at random among this set.

The pseudocode for the IPEG algorithm is provided in Table 2.2.

Table 2.2: Pseudocode for the IPEG Algorithm

For $j = 1$ to n

For $k = 1$ to $D_s(j)$

If $k == 0$

place edge between current VN v_j and CN c_i such that $c_i \in$ (the set of CNs with minimum weight under the current graph setting).

Else

expand tree to depth *l* under current setting s.t. either:

(1) the cardinality of $N_{v_j}^l$ stops increasing but is less than m

(2) $\overline{N_{v_j}^l} \neq \emptyset$ but $\overline{N_{v_j}^{l+1}} = \emptyset$

In the case of (1), place edge between current VN v_j and CN c_i s.t. $c_i \in \overline{N_{v_j}^l}$ with lowest CN degree. If a number of CN candidates meet this requirement, choose one at random.

In the case of (2), the set of minimum weight CNs of $\overline{N_{v_j}^l}$ is $\Omega_{v_j}^l$

If the cardinality of $\Omega_{v_j}^l == 1$

the check node $\Omega_{v_j}^l$ is connected to v_j

Else

For each CN $c_q \in \Omega_{v_j}^l$

calculate $ACE_{c_q} = \sum_{v_p}(w_{v_p} - 2)$ where the summation is taken over all VNs v_p in the cycle created by placement of edge connecting c_q to v_j.

End For

choose CN c_i s.t. $ACE_{c_i} \geq ACE_{c_q}$ for all $c_q \in \Omega_{v_j}^l$. If more than one CN meets this requirement, choose at random among the CNs which meet it.

End If

End If

End For

End For

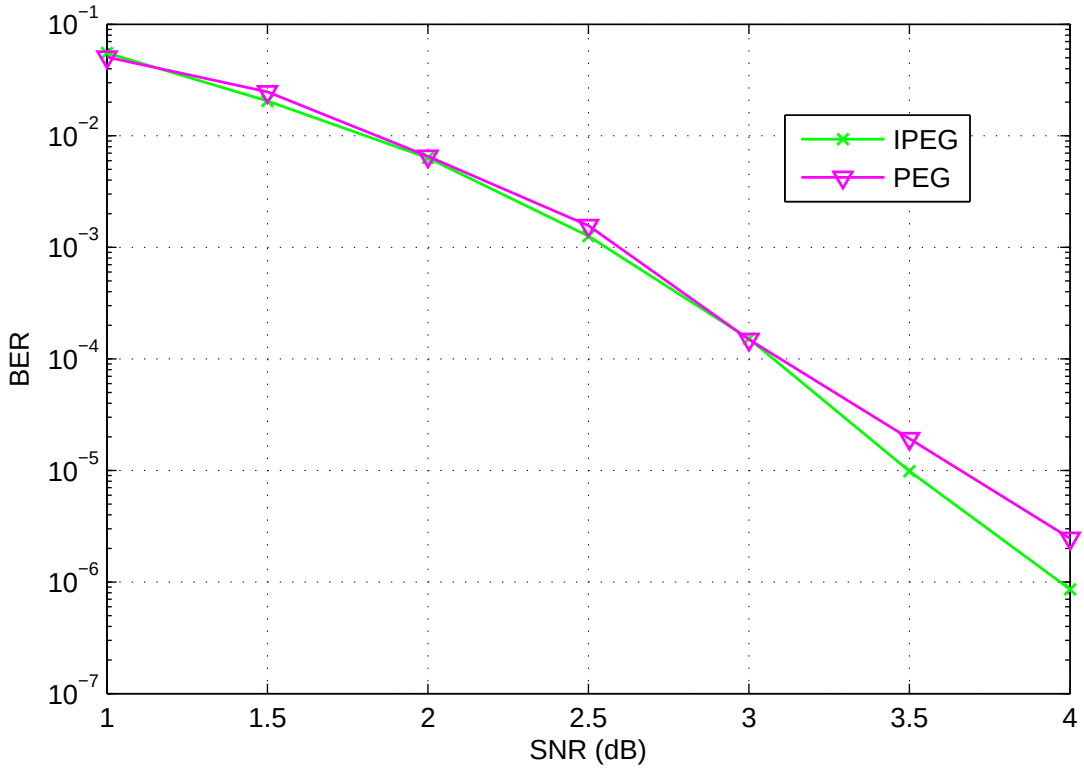


Figure 2.11: Comparison of Performance of IPEG and PEG generated codes

2.3.7 ACE Spectrum and ACE Constrained PEG Design

In [15] and [7] another approach to utilising the ACE concept of [8] is taken. As in the case of the IPEG, the motivation is that since stopping sets dictate the performance of LDPC codes in the error floor region, and since stopping sets are formed of one or more cycles, code performance can be improved by manipulating the cycles contained in the graph of the code. Again, as not all cycles are equally harmful, not only length but interconnectivity of cycles is considered.

In [15] a new metric based on ACE is introduced, the ACE spectrum of a graph

$$\eta\left(G(H)\right) = \left[\eta_2, \eta_4, \cdots, \eta_{2d_{\max}}\right], \tag{2.23}$$

where η_i is the minimum ACE of any cycle of length *i* in the graph. A construction method based on this graph metric is then proposed, whereby the paramaters $d_{\max}$ and η_{ACE} are set such that for any cycle of length less than or equal to $d_{\max}$ the ACE is

greater than or equal to η_{ACE}. This construction method produced codes which perform well, particularly when combined with the PEG algorithm in a similar approach to the IPEG modification of the PEG. It should be noted that there are limits for practical codes on the sizes of $d_{\max}$ and η_{ACE}, initially due to computational complexity and as they are increased it eventually becomes impossible to find graphs which comply.

In [7] in an extension of the work of [6] on the IPEG, the ACE spectrum as defined in [15] is used to identify a progression of culling of check node candidates as provided by the PEG algorithm. The minimum weight CN criterion is abandoned in favour of an ACE focused set of criteria. These are in order of sequence applied:

- select survivor with the largest minimum path ACE metric
- select survivor with the smallest number of minimum ACE shortest paths
- select survivor with the smallest total number of shortest paths
- select minimum degree survivor

2.4 Chapter Conclusions

In this chapter, a general introduction to the area of LDPC codes was provided. The coding system was presented and the iterative decoding procedure most commonly used was detailed. Along with these concepts, a number of more recent developements in the field were presented in the Literature Review section. This section provided an indication of the challenges faced in producing practical high performance codes and some of the approaches taken in meeting these challenges. Additionally, this section introduced a number of concepts, such as the PEG algorithm for PCM construction, which are key to the description of the proposed algorithm provided in the following chapter.

Plots of BER against SNR were provided for a number of the code design methods covered in order to give an appreciation of the performance achievable with LDPC codes at practical lengths.

Chapter 3

Proposed Decoder-Optimised PEG Algorithm

3.1 Introduction

As discussed when introducing the PEG algorithm for LDPC code construction, given a degree sequence derived from an optimal degree distribution and code paramaters, block length n and rate R, the PEG algorithm produces codes which exhibit excellent performance due to their girth characteristics. As demonstrated by the development of the IPEG construction method however, there is room for improvement of the PEG algorithm. In particular, it is noted in [6] that it regularly occurs that the PEG algorithm provides a number of candidate check nodes which are equivalent in terms of the length of the cycle which will be created if an edge is placed between them and the variable node of interest. The improvements in performance of the IPEG over the PEG algorithm results from calculating a metric, based on the approximate cycle EMD (ACE) of [8] for each candidate and by this means choosing the candidate with the greatest connectivity with the rest of the graph. This method was successful in producing codes with improved performance when compared with the original PEG algorithm.

An alternative approach is considered here, the decoder optimised PEG (DOPEG), which applies the SPA decoder at the design stage in an effort to produce an LDPC code with improved performance. By use of the decoder, a metric of comparison for candidate nodes is produced and the candidate is chosen based on this metric. As the results show, this approach leads to significant performance gains, particularly in the short to medium block length.

3.2 DOPEG Detailed Description

As in the PEG algorithm, placements are made edge-by-edge. Initialisation is identical to the PEG algorithm. For a variable node v_j the first edge is placed at random among the minimum weight check nodes of the graph. As in the PEG algorithm, a subtree is expanded from the variable node of interest to depth *l*, where the tree expanded to depth *l*+1 either (a) contains no more nodes than the tree to depth *l* or (b) contains all m check nodes of the graph. This implies that the nodes not included in the subtree at depth *l* are either unreachable from the variable node v_j or are at the greatest distance possible from this node. Up to this point the algorithm has been identical to the original PEG algorithm, as described in Section 2.3.5.

Now, if for the current variable node of interest v_j, the index j is less than the number of check nodes m, the candidate is chosen at random from the minimum weight check nodes not currently in the subtree i.e. the choice is made as in the original PEG algorithm. If the index is greater than the number of check nodes, in the case where there is more than one check node not currently in the subtree (in the case where there is only one node not in the subtree, this node is chosen and the algorithm moves on to the next edge to be placed) i.e. the cardinality of $\overline{N_{v_j}^l}$ is greater than 1, the elements of $\overline{N_{v_j}^l}$ are the candidate check nodes. Note that this differs from the PEG algorithm where the candidate check nodes are the minimum weight check nodes in the set $\overline{N_{v_j}^l}$ of nodes not in the tree to depth *l*.

Now for each candidate check node, a candidate code is constructed according to

the equations

$$\mathbf{H} = [\mathbf{I}_{\mathbf{n-k}}\ \mathbf{P}] \tag{3.1}$$

$$\mathbf{G} = [\mathbf{P}'\ \mathbf{I}_{\mathbf{k}}] \tag{3.2}$$

by Gaussian elimination, the method is discussed in Section 2.2.

For each candidate code, the LDPC coding system of Fig. 2.1 is operated over a range of signal-to-noise ratios (SNRs), for a number of input message vectors and in the presence of additive white Gaussian noise (AWGN) in the channel. The range of SNRs and number of instances of input message vectors are input parameters of the DOPEG algorithm.

For each candidate code the system consisting of encoding, transmission in the presence of AWGN, and soft-input soft-output SPA decoding is operated. The level of correct and incorrect convergence of the log-likelihood ratios (LLRs) of each bit in the soft output of the SPA decoder is measured.

This is then used to compute a single metric, as described in the section [3.3] to follow, for each candidate check node. The candidate producing the code with the highest metric, that is the candidate code which performs best under SPA decoding, is chosen as the candidate to connect to the current variable node of interest, v_j.

3.3 Metric Calculation

As described in the pseudocode of Table 3.1, for each candidate check node, encoding and soft-input/soft-output decoding is performed in the presence of additive white Gaussian noise over a range of SNR values and for a number of instances of message/noise vectors.

For each decoder soft-output vector, the magnitude of each bit log-likelihood ratio (LLR) is taken, and if the LLR is converging to the correct value, this magnitude is

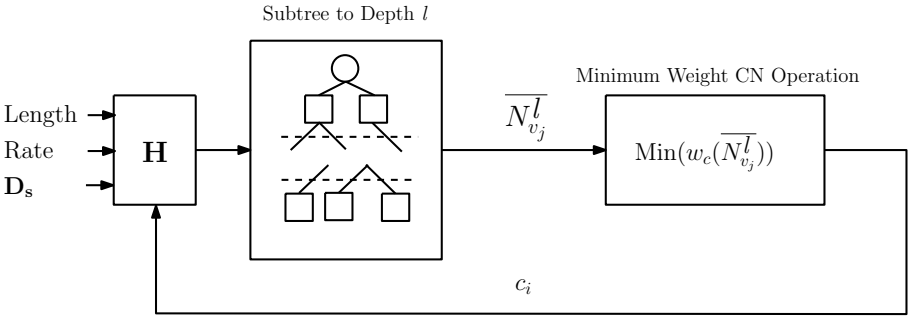
multiplied by +1, otherwise it is multiplied by -1. The entries of this new vector are then summed, and these sums are accumulated over the chosen number of instances of message/noise vectors.

This process is repeated at each SNR value for each candidate CN, resulting in a matrix of metric vectors for the CN candidates over the SNR range chosen. These metric vectors give an indication of how each candidate CN would affect the overall performance of the code. Rather than comparing the average of these metric vectors, which would fail to account for the greater convergence in SPA decoding at higher channel SNRs, the final metric for comparison is computed as follows: Taking the mean of the metrics at each SNR value over the different candidates, dividing each of the metrics at this SNR value by this mean value results in a normalised metric. This maintains the relationship of performance between different candidates at each SNR value of interest while removing the bias towards higher SNR values. These normalised metrics are then simply summed for each candidate, the largest value indicating the candidate with the best performance over the range of SNR values chosen.

3.4 Block Diagram and Pseudocode for the DOPEG Algorithm

Presented in Fig. 3.1 is a block diagram representation of the PEG and DOPEG algorithms for ease of comparison. In Table 3.1 the pseudocode for the DOPEG is presented.

PEG Algorithm



DOPEG Algorithm

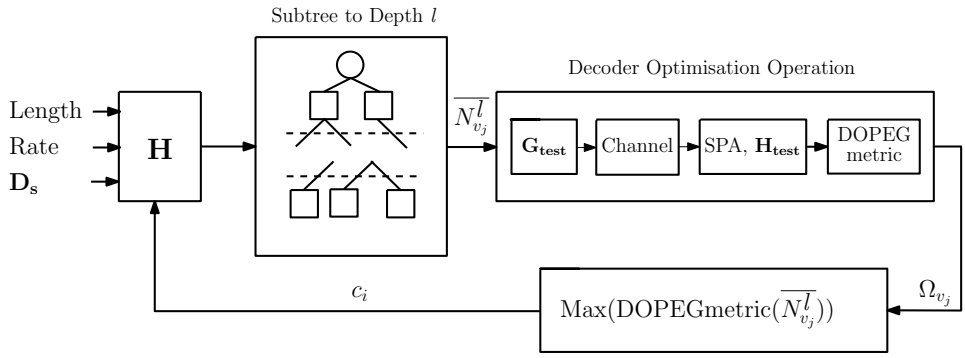


Figure 3.1: Block Diagram of PEG and DOPEG

Table 3.1: Pseudocode for the DOPEG Algorithm

For $j = 1$ to n

For $k = 1$ to $D_s(j)$

If $k == 0$

choose candidate at random from minimum weight CNs of the Tanner graph under the current graph setting.

Else

expand tree under current setting s.t. the cardinality of $N_{v_j}^l$ stops increasing but is less than m **or** $\overline{N_{v_j}^l} \neq \emptyset$ but $\overline{N_{v_j}^{l+1}} = \emptyset$

Then

If $j < m + 1$

choose candidate at random from minimum weight CNs among $\overline{N_{v_j}^l}$

Else

For $p = 1$ to length($\overline{N_{v_j}^l}$)

1. Form matrix $\mathbf{H_{test}}$ which is the constructed **H** matrix under the current graph setting up to column v_j, the current column of interest, with a 1 in the position [$\overline{N_{v_j}^l}$(p),v_j].

2. Use $\mathbf{H_{test}}$ to encode a message and decode in the presence of noise over a range of SNR values using soft-input soft-output log-domain SPA decoding.

3. Compute metric, described in Section (3.3), from the soft-output vectors of the SPA decoder.

End For

Choose the candidate with the highest metric, place edge in this position.

End If

End If

End For

End For

3.5 Decoder Optimised Improved PEG Algorithm

In an analogous fashion to the IPEG extension of the PEG algorithm, the DOIPEG combines the ACE metric concept as applied to the IPEG with the decoder optimisation step of the DOPEG algorithm. The set of check node candidates as presented by the PEG algorithm, $\overline{N_{v_j}^l}$, omitting the minimum weight check node stipulation as in the DOPEG, is pruned according to the ACE metric as defined for the IPEG. As a result, the surviving check node candidates have equal maximum graph connectivity as defined by the ACE metric of [8][6]. In the event that more than one check node remains in the set, the decoder optimisation procedure is carried out as in the DOPEG algorithm in order to provide a metric by which to choose the check node candidate which provides the best performance. This construction method is then called the Decoder Optimised Improved PEG (DOIPEG) algorithm. As the simulation results show, this leads to significant improvement in performance.

As in the DOPEG algorithm, the minimum check node requirement of the original PEG and IPEG algorithms is omitted. As such the resulting code will not have the concentrated check node degree distribution form of (2.12). Experimental results show that the removal of this stage of pruning of check node candidates leads to greater improvement in performance over the IPEG algorithm.

For the case where placement of an edge does not create a new cycle, i.e. either the first edge placed at a variable node or in the initial phase of graph construction when not all check nodes are reachable from the variable node of interest, the algorithm proceeds exactly as in the PEG algorithm.

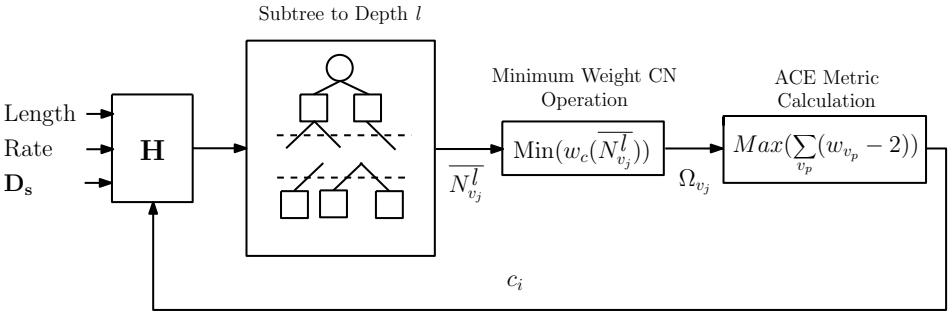
For the case where a cycle will be created with edge placement, a series of pruning operations are carried out on the set of check node candidates in order to identify the candidate which will provide best performance. The first pruning operation is that of the subtree expansion of the PEG algorithm which ensures that the cycle created will be of greatest length possible under the current graph settings. The second pruning operation is that of the IPEG algorithm, where the ACE metric is applied to ensure

that the survivnig check node candidates produce cycles of equal maximum graph connectivity. Finally, the decoder optimisation procedure described in sections (3.2) and (3.3) is carried out to identify which candidate will provide the best performance under encoding and SPA decoding.

As the simulation results show, the DOIPEG algorithm provides significant improvement in performance over the IPEG algorithm, which generates among the best performing codes currently known given an input degree sequence and rate.

A block diagram illustrating the approach of the DOIPEG algorithm is presented in Fig. 3.2. The block diagram for the IPEG algorithm is presented also for ease of comparison. The block diagrams of the PEG and DOPEG algorithms in Fig. 3.1 of Section 3.4 may also be useful for comparison between DOPEG and DOIPEG extensions of PEG and IPEG algorithms respectively. The pseudocode for the DOIPEG algorithm is provided in Table 3.2

IPEG Algorithm



DOIPEG Algorithm

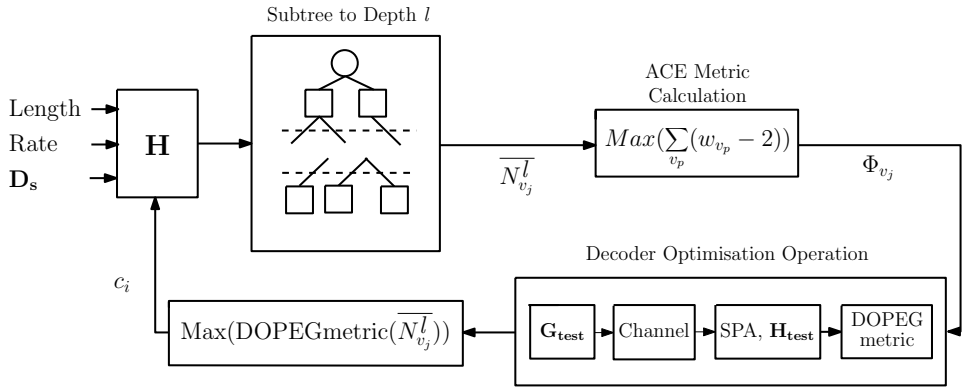


Figure 3.2: Block Diagram of IPEG and DOIPEG

Table 3.2: Pseudocode for the DOIPEG Algorithm

For $j = 1$ to n
For $k = 1$ to $D_s(j)$
If $k == 0$
place edge between current VN v_j and CN c_i such that $c_i \in$ (the set of CNs with minimum weight under the current graph setting).
Else
expand tree to depth *l* under current setting s.t. either:
(1) the cardinality of $N_{v_j}^l$ stops increasing but is less than m
(2) $\overline{N_{v_j}^l} \neq \emptyset$ but $\overline{N_{v_j}^{l+1}} = \emptyset$
In the case of (1), place edge between current VN v_j and CN c_i s.t. $c_i \in \overline{N_{v_j}^l}$ with lowest CN degree. If a number of CN candidates meet this requirement, choose one at random.
In the case of (2), the candidate CN set is now $\overline{N_{v_j}^l}$.
If the cardinaltiy of $\overline{N_{v_j}^l} == 1$
the check node $\overline{N_{v_j}^l}$ is connected to v_j
Else
For each CN $c_q \in \overline{N_{v_j}^l}$
calculate $ACE_{c_q} = \sum_{v_p}(w_{v_p} - 2)$ where the summation is taken over all VNs v_p in the cycle created placement of edge connecting c_q to v_j.
End For
then the set Φ_{v_j} is the set of CNs c_m s.t. $ACE_{c_m} \geq ACE_{c_q}$ for all $c_q \in \overline{N_{v_j}^l}$
If the cardinaltiy of $\Phi_{v_j} == 1$
the check node Φ_{v_j} is connected to v_j
Else
For $p = 1$ to length(Φ_{v_j})
[1] Form matrix $\mathbf{H_{test}}$ which is the constructed **H** matrix under the current graph setting up to column v_j, the current column of interest, with a 1 in the position [Φ_{v_j}(p),v_j].
[2] Use $\mathbf{H_{test}}$ to encode, decode over SNR range with SISO SPA decoder.
[3] Compute metric, described below, from the soft-output vectors of the SPA decoder.
End For
End If
End If
End If
End For
End For

3.6 Simulation Results

The simulation results consist of plots of BER vs signal-to-noise ratio (SNR) for codes generated by the PEG and DOPEG algorithms. In each case, codes of the same length have identical degree sequences, D_s, based on the variable node degree distribution

$$\lambda_1(x) = .30013x + .28395x^2 + .41592x^7 \tag{3.3}$$

The degree sequence was altered such that the number of weight-2 variable nodes is less than the number of check nodes. For PEG-based algorithms, this ensures that no cycles occur among only weight-2 variable nodes. The variable node degree distribution of (3.3) is density evolution optimised and was presented in [4] [Table I]. The codes are rate 1/2. For each plot, additional information is given below, in particular this specifies the parameters of SNR range, SPA decoder maximum iterations and number of message vectors generated, for which the Decoder Optimisation step was performed in the DOPEG algorithm.

For each of the plots in this section, BPSK modulation was used in the simulation, the transmitted symbols were subjected to AWGN and the SPA decoder was used in the receiver. In simulating the coding system, for length $n = 250$ codes the SPA decoder was operated to a maximum of 40 iterations and 100 block errors were gathered for each point in the BER curves. For $n = 500$ codes the SPA decoder was operated to a maximum of 10 iterations and 100 block errors were gathered for each point.

3.6.1 DOPEG

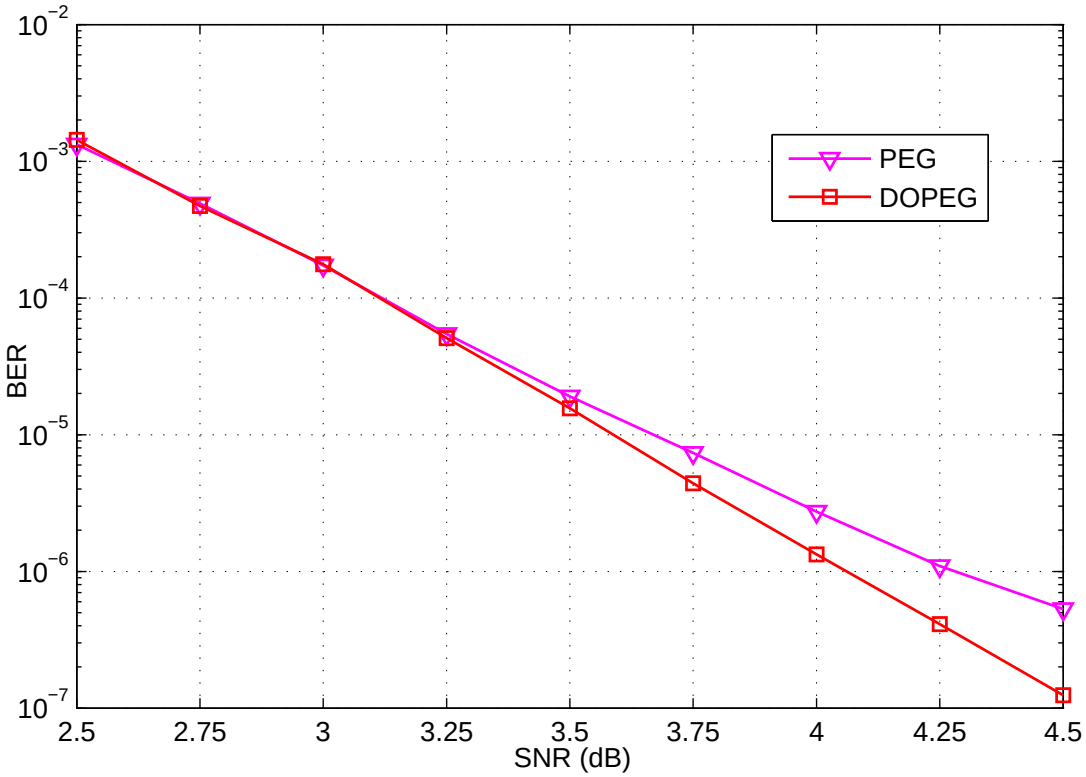


Figure 3.3: Comparison of Performance of DOPEG and PEG generated codes for length $n = 250$

In Fig. 3.3 above, the BER curves of the PEG algorithm and the DOPEG algorithm constructed codes are compared. For the DOPEG, the decoder optimisation was carried out over the SNR range [1:0.05:2], with 5 instances of message vectors generated and the SPA decoder was operated to the maximum number of 50 decoder iterations.

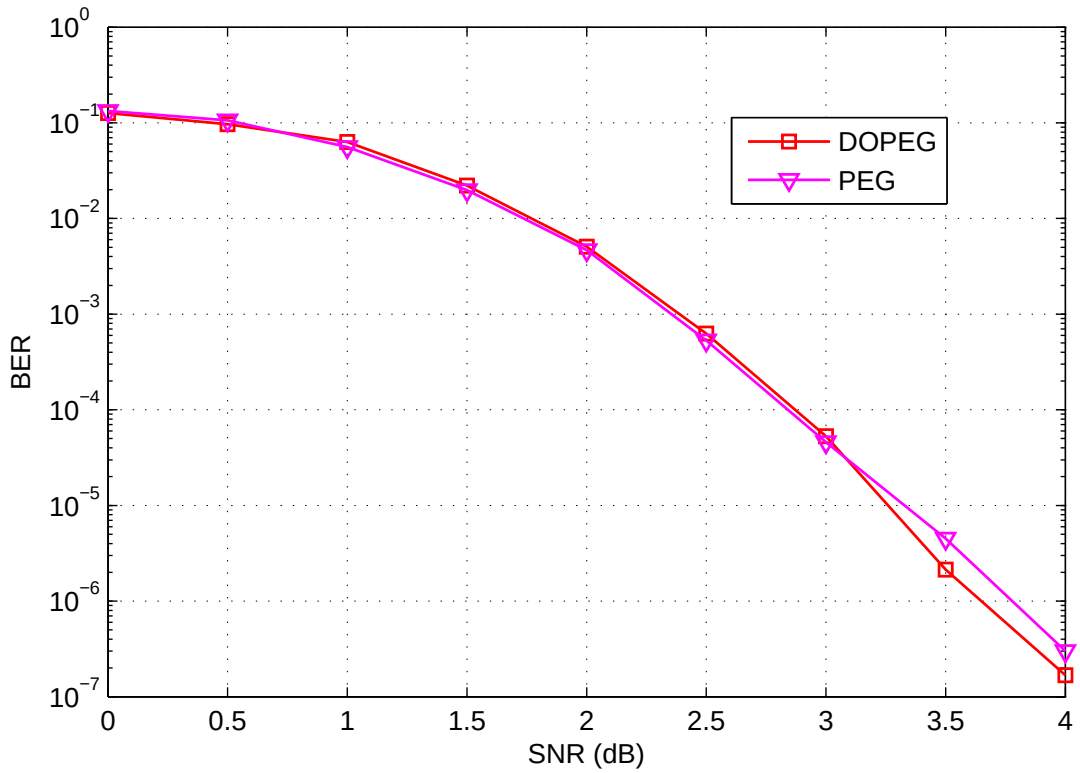


Figure 3.4: Comparison of Performance of DOPEG and PEG generated codes for length $n = 500$

The results above were found for the DOPEG generated code with block length 500, rate $\frac{1}{2}$ and variable node degree distribution described by Eqn. 3.3 with maximum variable node degree 8. The parity-check matrix for the DOPEG code was generated for the decoder optimisation procedure operating over the SNR range [1:0.05:3] and with 60 instances of message vectors generated for each candidate check node. The decoder in the DO stage was operated at 50 decoder iterations.

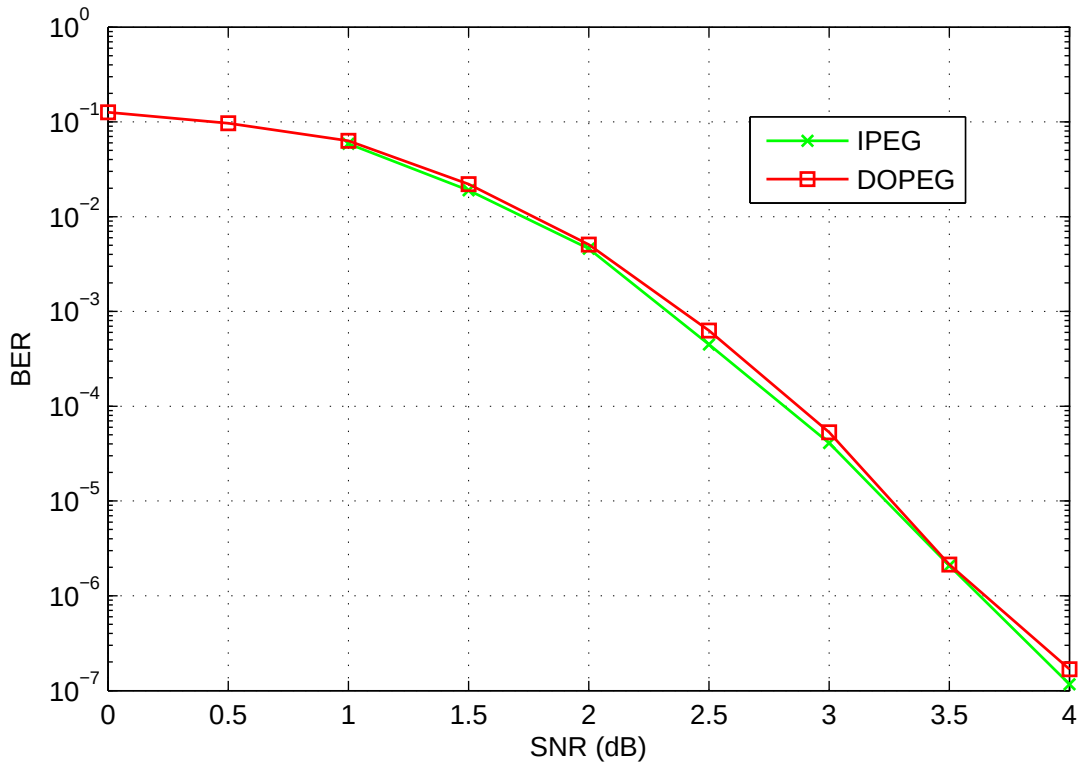


Figure 3.5: BER vs SNR length $n = 500$ DOPEG and IPEG constructed LDPC codes

In Fig. 3.5 above, the performance of the length $n = 500$ rate 1/2 irregular DOPEG constructed code with maximum variable node degree distribution 8 is compared to that of the IPEG constructed code with identical length and degree distribution. The decoder optimisation step was carried out for the SNR range [1:0.05:3] and 60 instances of message vectors where generated for each candidate CN. The decoder in the DO stage was operated at 50 iterations.

3.6.2 DOIPEG

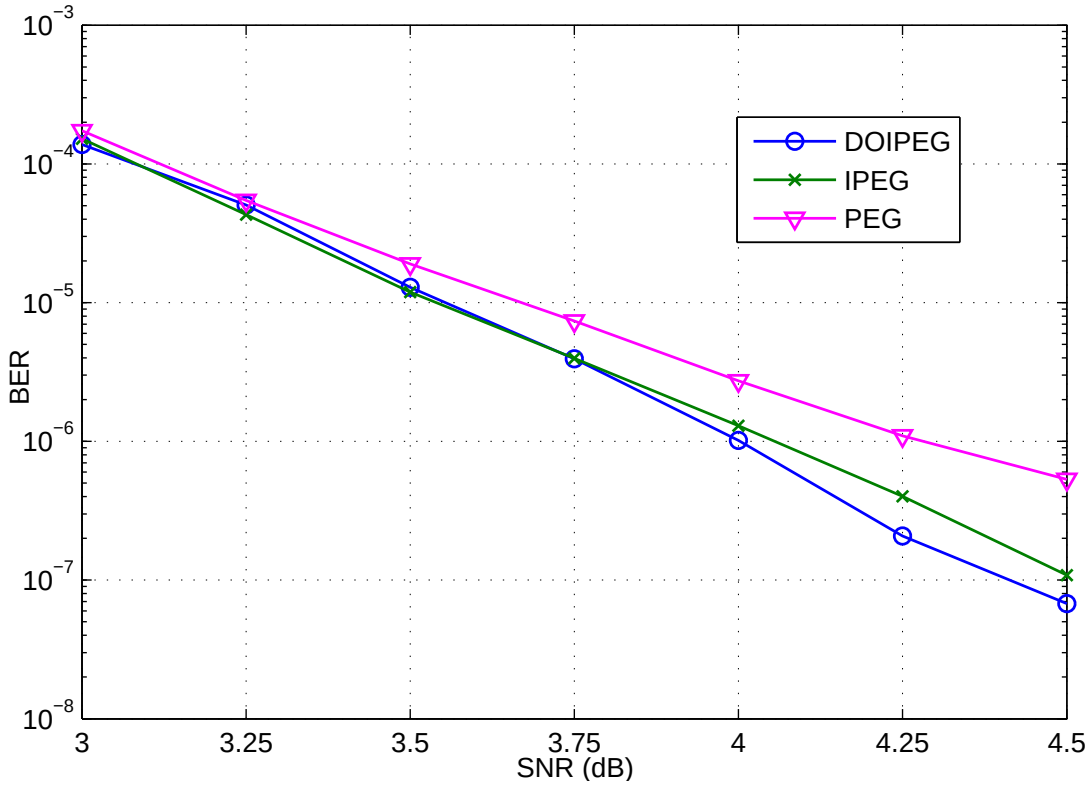


Figure 3.6: Comparison of Performance of DOIPEG, IPEG and PEG generated codes for length $n = 250$

The results above were found for the DOIPEG generated code with block length 250, rate $\frac{1}{2}$ and variable node degree distribution from [4] Table I with maximum variable node degree distribution 8. The parity-check matrix was generated by the DOIPEG with the decoder optimisation step operating over the SNR range [1:0.05:2] and with 5 instances of message vectors generated for each candidate check node.

The gain exhibited by the DOIPEG over the IPEG is less dramatic than that of the DOPEG over the PEG codes of Fig. 3.3. This is due to the fact that much of the benefit of the DOPEG over the PEG is shared with the IPEG. That is, a candidate edge which is optimal in terms of the ACE metric, and therefore is chosen by the

IPEG algorithm is more likely to result is better performance under SPA decoding in the decoder optimisation (DO) step and vice versa. However, as graph connectivity as measured by the ACE metric (itself an approximation to the EMD of a node) alone does not wholly dictate performance of an LDPC code in the AWGN channel improvement upon the IPEG code is possible, given a large enough set of message vectors in the DO step. This is the reason for the large number of message vectors used in this example compared to that of Fig. 3.3.

3.7 Chapter Conclusions

In this chapter, the proposed algorithm for LDPC parity-check matrix construction, which forms the core of the contributions made in this thesis, was described and its performance was analysed. The approach taken was described in detail and a block diagram was included which provides an overview of the steps taken by the construction method and which details how it differs from the original PEG algorithm. Pseudocode was also provided to give a more detailed view of the algorithm.

The DOIPEG extension of the DOPEG algorithm was then presented. This extension is analogous to the IPEG extension to the original PEG algorithm. LDPC codes with optimum degree distribution as found by density evolution and constructed using the IPEG algorithm are among the best performing codes currently available and so improvement upon their performance is noteworthy.

In Section 3.6 the performance of the proposed codes was investigated and compared to the performance of the PEG and IPEG generated codes. As discussed, the performance results achieved are particularly significant given that all extra effort in terms of computation is exacted at the design stage and during transmission the generated codes are equivalent to PEG and IPEG generated codes in encoding and decoding complexity.

Chapter 4

Iterative Detection and Decoding of MIMO Systems with LDPC Codes

In this chapter we review MIMO systems. Following a general introduction, a number of detection algorithms are described. An LDPC coded iteratively detected and decoded spatially multiplexed MIMO system is then reviewed and a soft-in soft-out (SISO) detector for use in this system is detailed.

Through simulation results, the operation and performance of this system is shown. Then the DOPEG developed in Chapter 3 is applied to the system and its performance is compared to that of the turbo (iterative) detected and decoded MIMO system with PEG coding.

4.1 An Introduction to MIMO Systems

A multiple-input multiple-output (MIMO) system, as shown in Fig. 4.1, has N_t transmit antennas and N_r receive antennas [26][27]. We are interested here in a spatial multiplexing configuration, where independent signals are transmitted from each of the N_t transmit antennas leading to a multiplexing gain of N_t over the single-input sinlge-output system [26]. Transmission in the system is then described by the equa-

tion

$$\mathbf{r} = \mathbf{Hs} + \mathbf{n}, \tag{4.1}$$

where **s** is the $(1 \times N_t)$ vector of information symbols to be transmitted and **r** is the $(1 \times N_r)$ vector of received symbols. The $(1 \times N_r)$ vector **n** is the noise vector. **H** is, for the case of a flat fading channel, the $(N_r \times N_t)$ matrix which describes the paths between each transmit and receive antenna, as shown below

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1N_t} \\ h_{21} & h_{22} & \cdots & h_{2N_t} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_r1} & h_{N_r2} & \cdots & h_{N_rN_t} \end{bmatrix} \tag{4.2}$$

where h_{ij} is the complex zero-mean Gaussian channel-fading coefficient for the path from the *j*th transmit antenna to the *i*th receive antenna. Then the signal received at the *i*th receive antenna is described by the equation

$$r_i = \sum_{j=1}^{N_t} h_{ij} s_j + n_i \tag{4.3}$$

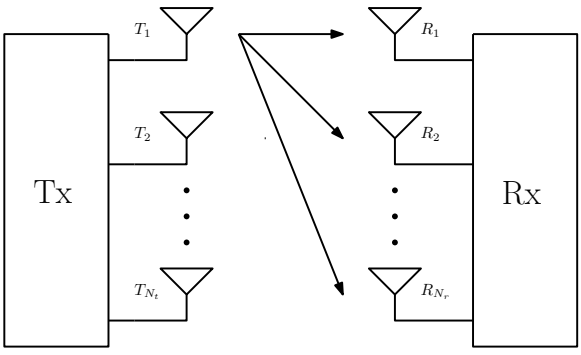


Figure 4.1: Block Diagram of a General MIMO System

4.2 Detection Algorithms for MIMO Systems

In a spatial multiplexing system, the data stream to be transmitted is demultiplexed, modulated and transmitted over N_t transmit antennas in parallel [26]. The signals are transmitted in the same frequency band and so at each receive antenna the superposition of all transmit signals, degraded according to the path from each transmit to receive antenna, is received. These N_r receive signals are subject to AWGN at each receive antenna also. The challenge of MIMO detection is to separate out and recover each signal which was transmitted, mitigating the effects of the co-channel interference and noise.

The optimum maximum likelihood (ML) receiver accomplishes this by exhaustive search over all possible transmitted signals [27]. However this is in general far too complex an approach for practical use [28]. Consequently lower complexity detection schemes have been developed which result in some sacrifice in performance with respect to ML detection. An so the goal, ultimately, is to find an acceptable tradeoff between computational complexity and performance.

4.2.1 Maximum Likelihood Detection

The optimum receiver for a MIMO system uses the Maximum Likelihood Detector (MLD) which performs an exhaustive search over all possible transmitted symbols in order to minimise the probability of error. The MLD solves

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \|\mathbf{r} - \mathbf{H}\mathbf{s}\|^2 \tag{4.4}$$

where $\hat{\mathbf{s}}$ is the estimated symbol vector. The complexity of the MLD grows exponentially with the number of transmit antennas and the number of points in the signal constellation, and as such is too complex for practical implementation. However, algorithms have been developed which approach the performance of the MLD with reduced complexity, one such detector is called the Sphere Decoder [29][30]. The Sphere De-

coder provides a tradeoff between complexity and performance, with both being highly sensitive to the choice of the sphere radius. The Sphere Decoder is attractive for small systems, however, its complexity scales in an exponential form with the number of data streams [31].

4.2.2 Linear Minimum Mean-Squared Error Detection

The linear minimun mean square error (MMSE) detector minimises the overall error due to the combined factors of noise and mutual interference of co-channel signals. This is achieved by minimising the mean square error

$$\mathrm{MSE} = E\left[\|\mathbf{s} - \mathbf{W}^H\mathbf{r}\|^2\right] \tag{4.5}$$

Practically, linear MMSE detection is achieved by multiplying the received vector **r** by the complex conjugate of the $N_r \times N_t$ weighting matrix **W** to find the estimate of the transmitted vector as

$$\hat{\mathbf{s}} = \mathbf{W}^H\mathbf{r} \tag{4.6}$$

where the weighting matrix **W** is

$$\mathbf{W} = \left(\frac{1}{\mathrm{SNR}}\mathbf{I}_{\mathbf{N_r}} + \mathbf{H}\mathbf{H}^H\right)^{-1}\mathbf{H}^H \tag{4.7}$$

and the superscript H denotes the complex conjugate transpose. The computational complexity of the MMSE detector grows as a cubic function of N_r for the matrix inversion required in Eqn. 4.7 and as a function of N_r times N_t for the filtering operation in Eqn. 4.6.

4.2.3 V-BLAST Detection

The Vertical Bell Labs Layered Space Time (V-BLAST) detector employs successive interference cancellation (SIC) to yield improved performance at the cost of increased

complexity over the linear detector [9][32]. This detection algorithm operates in an iterative fashion, first detecting the strongest substream of the received signals and then proceeding to the weaker substreams, which are then easier to detect as the stronger signals are subtracted and no longer provide a source of interference. The V-BLAST algorithm carries out nulling, slicing and cancellation steps according to a chosen ordering. In the literature [32] it has been reported that an ordering which starts with the strongest signal and proceeds to the weakest signal provides the best performance. The algorithm may be summarised as follows:

Given the initial received vector

$$\mathbf{r_1} = \mathbf{Hs} + \mathbf{n} \tag{4.8}$$

Step 1: Use the vector $\mathbf{w_{k_1}}$, the nulling vector to produce an estimate of the strongest transmitted signal by nulling out the weaker transmit signals

$$y_{k_1} = \mathbf{w_{k_1}^T r_1} \tag{4.9}$$

Step 2: Slice this transmit signal estimate according to the appropriate operation for the constellation used in order to produce an estimate of the symbol transmitted

$$\hat{s}_{k_1} = Q(y_{k_1}) \tag{4.10}$$

Step 3: This estimate of the symbol transmitted is applied to the channel in order to estimate its contribution to the received vector. This is then cancelled from the received vector, thus removing the interference provided by this transmit substream. This is carried out as

$$\mathbf{r_2} = \mathbf{r_1} - (\mathbf{H})_{\mathbf{k_1}} \hat{\mathbf{s}}_{\mathbf{k_1}} \tag{4.11}$$

where $(\mathbf{H})_{\mathbf{k_1}}$ is the k_1-th column of $\mathbf{H}$.

These steps are repeated in an iterative fashion until each of the N_t transmit substreams have been detected. The specifics of the nulling step, that is the criterion for

choosing the nulling vectors $\mathbf{w_{ki}}$ provides some flexibility. The most common choices for this criterion are MMSE and zero forcing (ZF) [26]. An example of MMSE-SIC detection may be found in [33].

4.2.4 Other Detectors

A number of other detection schemes exist which attempt to provide acceptable approximations to the optimal performance offered by the maximum likelihood detector while offering more practical levels of complexity.

Decision Feedback

The MMSE decision feedback equiliser was originally developed for the single-input single-output system to tackle inter-symbol interference (ISI) by using previously detected symbols to cancel their interference contribution to the received signal at the current time. This approach was applied to the case of a MIMO system and combined with successive interference cancellation [34][35]

Parallel Interference Cancellation

In parallel interference cancellation detection, after initial conventional detection the individual streams are detected in parallel. For each stream of interest, the interfering signals due to all other streams are reconstructed using the channel matrix and subtracted from the received signal. [36]. This has also been combined with decision feedback [37].

Lattice Reduction Aided Techniques

In contrast to the previous schemes which employ techniques to progressively improve the detected symbol and which result in high complexity when compared to LMMSE detection, the approach taken in lattice reduction is to perform a single computationally costly operation at the start of a frame, followed by simple low-complexity de-

tection. This detection may be, for example, linear MMSE or SIC detection. This is accomplished by transforming the system model into an equivalent with a better conditioned channel matrix, then employing the lower-complexity detector. This results in improved performance over a system using the same detector in the presence of an ill-conditioned channel matrix [38].

4.3 The Iterative Detection and Decoding Principle for MIMO Systems

The Turbo Principle of decoding a serially concatenated encoded bit stream by soft inner and outer decoders exchanging iteratively updated extrinsic information in order to increase performance was first introduced by Berrou et al. [39]. It may be applied to the problem of detection and decoding of a MIMO system. In the work by Wang and Poor [28], the turbo principle was applied to the case of coded CDMA. In [40] this strategy is applied to a MIMO system with LDPC coding and a number of reduced complexity detectors are presented. When the turbo principle is applied to a MIMO system the channel decoder, here the SPA decoder for decoding an LDPC code, is viewed as the outer code, while the soft-input soft-output (SISO) MIMO detector is viewed as the inner code.

The block diagram of the LDPC-coded iteratively detected and decoded MIMO system is presented in Fig. 4.2. The information bits are first encoded by the LDPC encoder, as in Eqn. 2.1, then the encoded bits are interleaved, demultiplexed into N_t bit streams, each stream is modulated using the appropriate modulation scheme and then transmitted over its corresponding transmit antenna.

At the receiver, the signal is received at each of the N_r receive antennas. The SISO detector, operating in an iterative fashion, exchanging extrinsic information with the SISO SPA decoder and incorporating the information provided into the detection scheme used in order to improve its performance. A number of detection schemes exist

which provide capacity-approaching performance with varying degrees of complexity, for example the optimal but complex MAP detector, the MMSE Successive Interference Cancellation (MMSE-SIC) detector and the MMSE Hard Interference Cancellation (MMSE-SIC) detector.

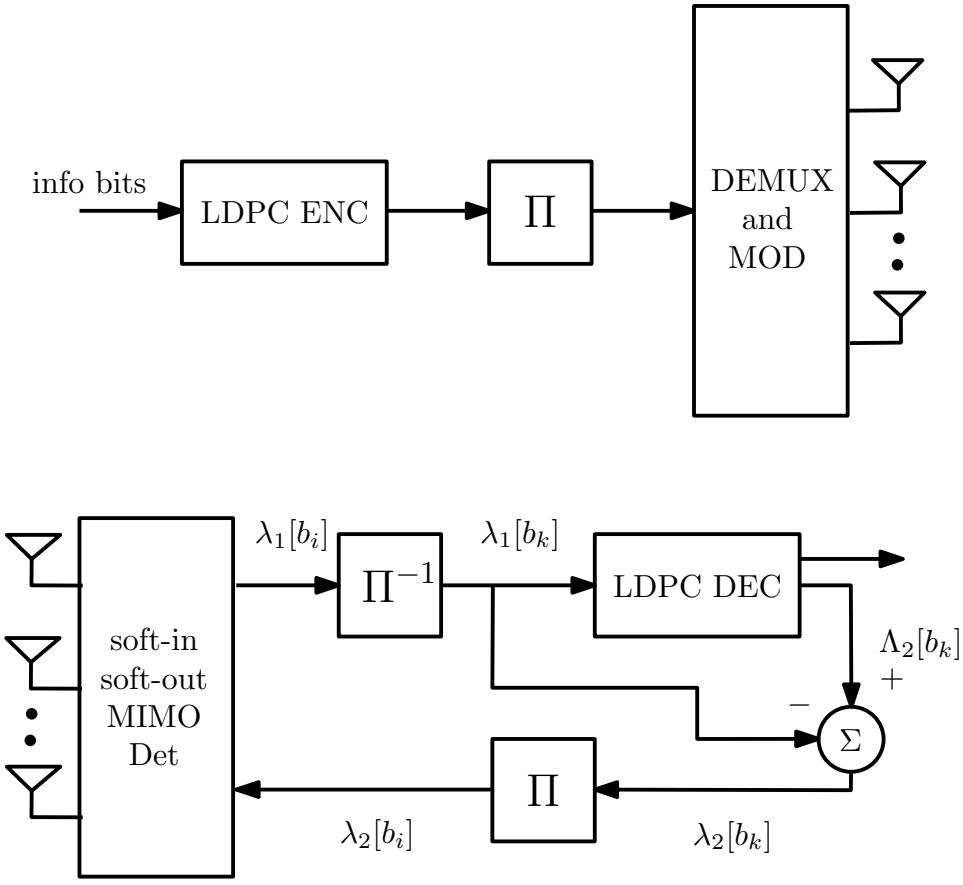


Figure 4.2: Iteratively Detected and Decoded MIMO System

4.3.1 Iterative Detection and Decoding Procedure

First Half-Iteration

Based on the iterative (turbo) multiuser receiver structure of Wang and Poor [28], the receiver structure used treats the signals transmitted from each transmit antenna as separate users at the detection stage. It follows then from [28] that the soft-input soft-output (SISO) detector computes the *a posteriori* log-likelihood ratio (LLR) of each of the transmitted coded bits, giving a measure of the probability that each bit was transmitted as a "+1" or a "-1". That is

$$\Lambda_1[b(i)] = \log\frac{p(b(i)=+1|\mathbf{r})}{p(b(i)=-1|\mathbf{r})}, \tag{4.12}$$

for each $i = 1, \cdots, n$

Where n is the block length of the LDPC code.

By Bayes' Rule, this is rewritten as

$$\Lambda_1[b(i)] = \log\frac{p(\mathbf{r}|b(i)=+1)}{p(\mathbf{r}|b(i)=-1)} + \log\frac{P[b(i)=+1]}{P[b(i)=-1]}, \tag{4.13}$$

where the first term is taken to be

$$\lambda_1[b(i)] = \log\frac{p(\mathbf{r}|b(i)=+1)}{p(\mathbf{r}|b(i)=-1)} \tag{4.14}$$

and the second term is taken to be

$$\lambda_2^p[b(i)] = \log\frac{P[b(i)=+1]}{P[b(i)=-1]}. \tag{4.15}$$

The quantity $\lambda_1[b(i)]$ is the extrinsic information which is to be passed to the channel decoder, here the SPA decoder, for use in the second half-iteration of the turbo detection/decoding procedure. The term $\lambda_2^p[b(i)]$ is the *a priori* LLR of the coded bit $b(i)$, received from the channel decoder in the previous iteration, as is indicated by the

p superscript. This *a priori* LLR is subtracted from the *a posteriori* LLR in order to maintain independence of the messages passed. At the first iteration, from the assumption that the coded bits are equally likely to be "+1" and "-1" the *a priori* LLR is set to zero.

Second Half-Iteration

The extrinsic LLR $\lambda_1[b(i)]$ is deinterleaved before being fed into the SPA decoder as *a priori* information. The SPA decoder operates precisely as in Section 2.2.3 in Fig. 2.3, iteratively operating on the graph structure of the code to produce the soft-output *a posteriori* LLR of each coded bit:

$$\Lambda_2[b(k)] = \log\frac{p(b(k)=+1|\lambda_1^p[b(l)]_{l=1}^{n-1},\mathbf{H})}{p(b(k)=-1|\lambda_1^p[b(l)]_{l=1}^{n-1},\mathbf{H})} \quad \text{for } k=1,\cdots,n-1 \tag{4.16}$$

Now

$$\Lambda_2[b(k)] = \lambda_2[b(k)] + \lambda_1^p[b(k)] \tag{4.17}$$

Again, the *a priori* probability, now $\lambda_1^p[b(k)]$ from the first half-iteration, is subtracted from the *a posteriori* LLR to produce the extrinsic information $\lambda_2[b(k)]$. This will be interleaved and fed back into the SISO detector for use in the next iteration.

In Fig. 4.2 the structure of the turbo detector and receiver is shown and the messages passed are indicated. As above the lower case letters indicate extrinsic information while the upper case indicates a posteriori information.

4.3.2 SISO MMSE Successive Interference Cancellation (SIC) Detector

The minimum mean square error successive interference cancellation (MMSE-SIC) detector for turbo detection and estimation was presented in [28] as a SISO multiuser

detector for use in a coded CDMA (code division multiple access) system. The detector here is presented for the case of a flat fading channel and so intersymbol interference (ISI) is not an issue. The challenge of detection in this scenario is to mitigate the effects of interference between the antennas of the system. This is the scenario which was considered for the system analysed with LDPC coding in this thesis. A more general approach is taken in [41] where a frequency-selective channel model is considered. As such both ISI and self-interference are encountered. The MMSE-SIC detector is developed in the absence of self-interference in [42], i.e., for a system with a single transmit antenna, for a frequency selective channel. Of particular interest considering the system of interest here is the work by Wesel et al. [40] where the frequency-nonselective channel is assumed and a number of detectors are developed for use in an iterative (turbo) detected and decoded MIMO system with LDPC coding.

Using the *a priori* LLRs of all coded bits provided by the channel decoder as extrinsic information, the MMSE detector forms soft symbol estimates of the bits transmitted from the *p*th transmit antenna as

$$\tilde{b_p}(i) = \tanh\left[\frac{\lambda_2[b_p(i)]}{2}\right] \quad \text{for } p = 1, \cdots, N_t \tag{4.18}$$

These estimates are then assembled into a replica, $\tilde{\mathbf{s}}$ and used to perform soft interference cancellation of the interference between antennas.

$$\hat{\mathbf{s}} = \mathbf{r} - \mathbf{H}\,\tilde{\mathbf{s}} \tag{4.19}$$

At this point, $\hat{\mathbf{s}}$ is our estimate of the transmitted signal. However, as our soft symbol estimates are not completely accurate, interference residuals exist. In order to mitigate their effect, adaptive filtering is carried out to suppress the residuals. The filter $\mathbf{w_p}$ is chosen to minimise the mean square error (MSE) between the transmitted bit $b_p(i)$ and the filter output $z_p(i)$. That is

$$\mathbf{w_p}(i) = \underset{w_p(i)}{\arg\min}\, E\left[\|b_p(i) - w_p^H(i)\hat{s}(i)\|^2\right] \tag{4.20}$$

The result of this minimisation is shown to be [28]

$$\mathbf{w_p}(i) = \left[\sigma^2\mathbf{I_{n_r}} + \mathbf{H}\ \mathbf{\Lambda_p}\ \mathbf{H}^H\right]^{-1}\mathbf{h_p} \tag{4.21}$$

where $\mathbf{h_p}$ is the p-th column of $\mathbf{H}$ and the matrix $\mathbf{\Lambda_p}$ is the covariance matrix

$$\mathbf{\Lambda_p} = \text{diag}\left[(1\ -\ \tilde{b_1}(i))\cdots 0\cdots(1\ -\ \tilde{b_{N_t}}(i))\right] \tag{4.22}$$

Where the zero is in the *p*th position. Now the filter output is

$$z_p(i) = \mathbf{w_p^H}\ \hat{\mathbf{s}} \tag{4.23}$$

Approximating the soft filter output as a Gaussian process

$$z_p(i) = \mu_p(i)b_p(i) + \nu_p(i) \tag{4.24}$$

as in [28] the information to be delivered to the channel decoder can be found as

$$\lambda_1[b_p(i)] = \frac{4\ Re[z_p(i)]}{1\ -\ \mu_p(i)} \tag{4.25}$$

Note

Two particular modes of operation of the MMSE-SIC detector are noteworthy

- No Cancellation: This occurs at the first iteration when no information is available from the decoder. The MMSE-SIC filter in this case reduces to the form of the linear MMSE filter of Eqn. 4.7 with $\mathbf{w_p}$ rather than $\mathbf{W}$.
- Perfect Cancellation: the MMSE-SIC filter reduces to the form

$$\mathbf{w_p} = \left(\frac{1}{\text{SNR}} + \mathbf{h_p^H h_p}\right)^{-1}\mathbf{h_p} \tag{4.26}$$

MMSE Hard Interference Cancellation Detector

Also presented in [40] is the MMSE hard interference cancellation (MMSE-HIC) detector which sacrifices performance for lower complexity. Here, after a prescribed number of iterations to ensure reliability of the information available, a hard decision is made about the value of the bits before the interference cancellation operation is carried out. This hard decision is made based on information available from the channel decoder.

4.4 Simulation Results

For the simulation results presented, BPSK modulation was employed. The transmitted signals were subjected to flat Rayleigh fading in the channel and corrupted by AWGN with zero mean and variance σ_n^2. For each plot which follows, an accompanying section of text provides information necessary in order to make useful comparisons of performance of the system under the prescribed settings. These settings include detector used and for the iterative detected and decoded system include the number of outer (detector) iterations and inner (decoder) iterations carried out and the LDPC code used. Following this, an analysis of each result is made.

A Note on SNR Calculation for the MIMO System Model Used

The noise variance of the AWGN of Eqn. 4.1 is calculated according to, for the uncoded system

$$\mathrm{SNR} = N_t \frac{\sigma_s^2}{\sigma_n^2}, \tag{4.27}$$

where σ_s^2 is signal variance per transmit antenna and σ_n^2 is the vector noise variance. For the coded system this becomes

$$\mathrm{SNR} = N_t R_c \frac{\sigma_s^2}{\sigma_n^2}, \tag{4.28}$$

where R_c is the code rate.

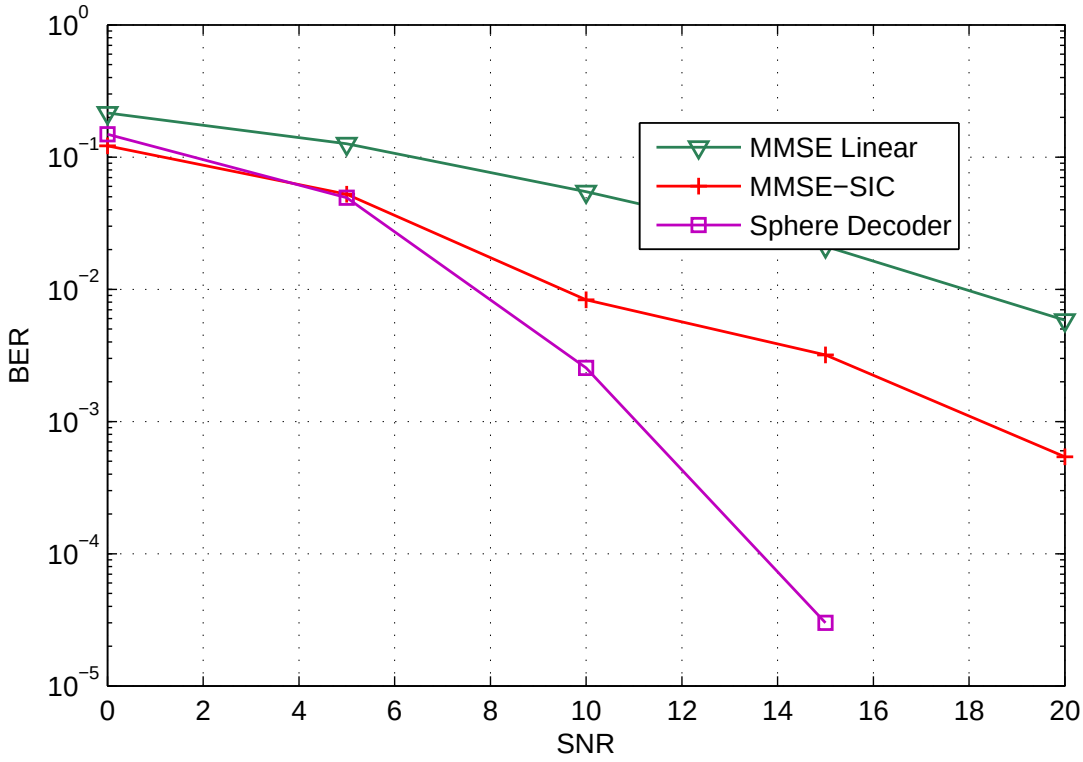


Figure 4.3: BER vs SNR comparison of different detectors for an uncoded 4x4 MIMO system

In the plot in Fig. 4.3 the performance of a number of detectors for a 4x4 uncoded spatially multiplexed MIMO system is shown. The detectors used were Sphere Decoder (SD), minimum mean square error successive interference cancellation (MMSE-SIC) and a linear MMSE detector. As expected from the theory the sphere decoder provides the best performance as it approximates the optimal maximum likelihood detector, while the linear MMSE detector provides the worst performance.

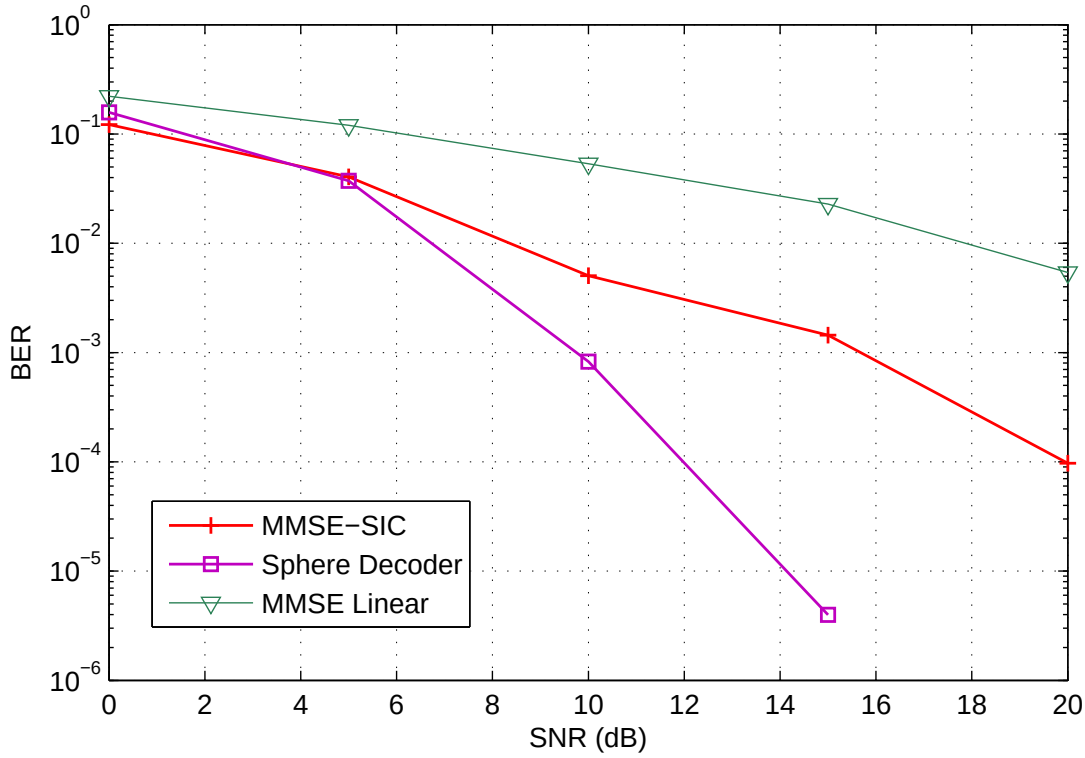


Figure 4.4: BER vs SNR comparison of different detectors for an uncoded 6x6 MIMO system

Fig. 4.4 provides a similar comparison to that of Fig. 4.3 but for the case of a 6x6 uncoded spatially multiplexed MIMO system. As expected, the results are similar.

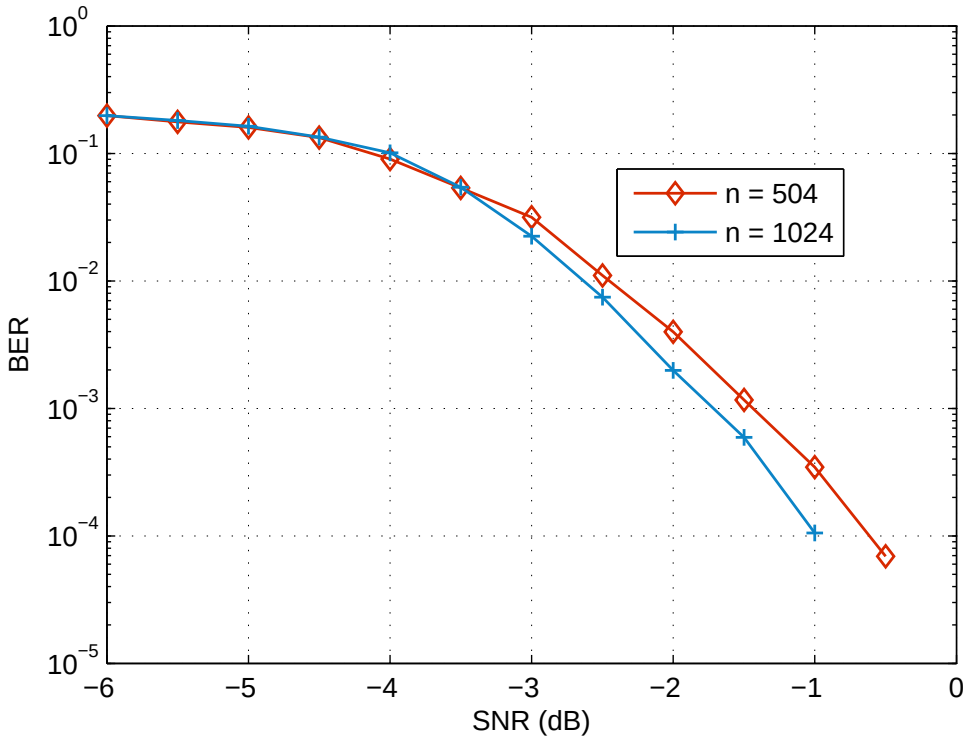


Figure 4.5: Comparison of BER performance of coded turbo 8x8 MIMO system with PEG codes of length $n = 504$ and $n = 1024$

The turbo (iterative) detector and decoder were operated for 5 iterations in the outer loop and the channel decoder was operated for 50 iterations. 100 block errors were accumulated per point of the plot. As is expected, as the length of the LDPC code used is increased performance increases. The codes used are PEG constructed and as such, the minimum distance of the code grows linearly with number of checks and code performance is highly dependent on the minimum distance of the code. The precise bound on minimum distance for PEG codes may be found in [5].

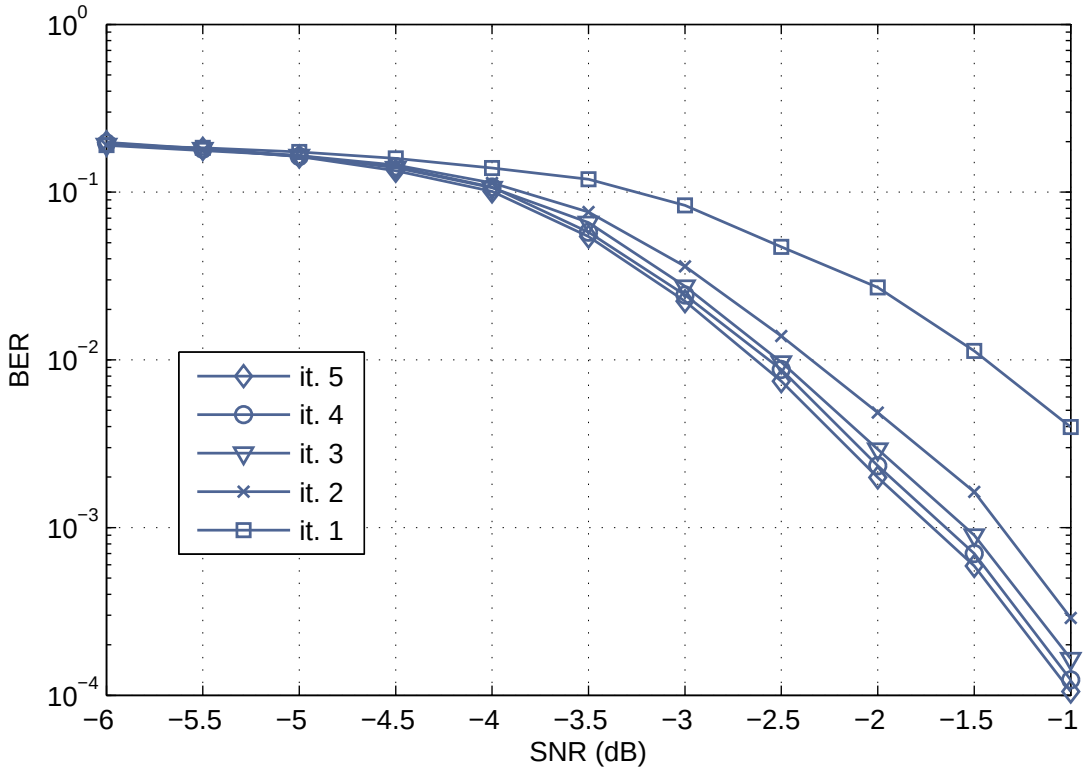


Figure 4.6: BER vs SNR for length $n = 1024$ irregular LDPC code operating over an 8x8 MIMO system showing the improvement in performance as iterations proceed

For Fig. 4.6 the turbo detector was operated for the same parameters as for the previous Fig. 4.5. This plot demonstrates the performance improvements as the joint detector and decoder move through the outer loop iterations. As ecpected, the performance improvement gained per extra iteration diminishes as the iteration number increases.

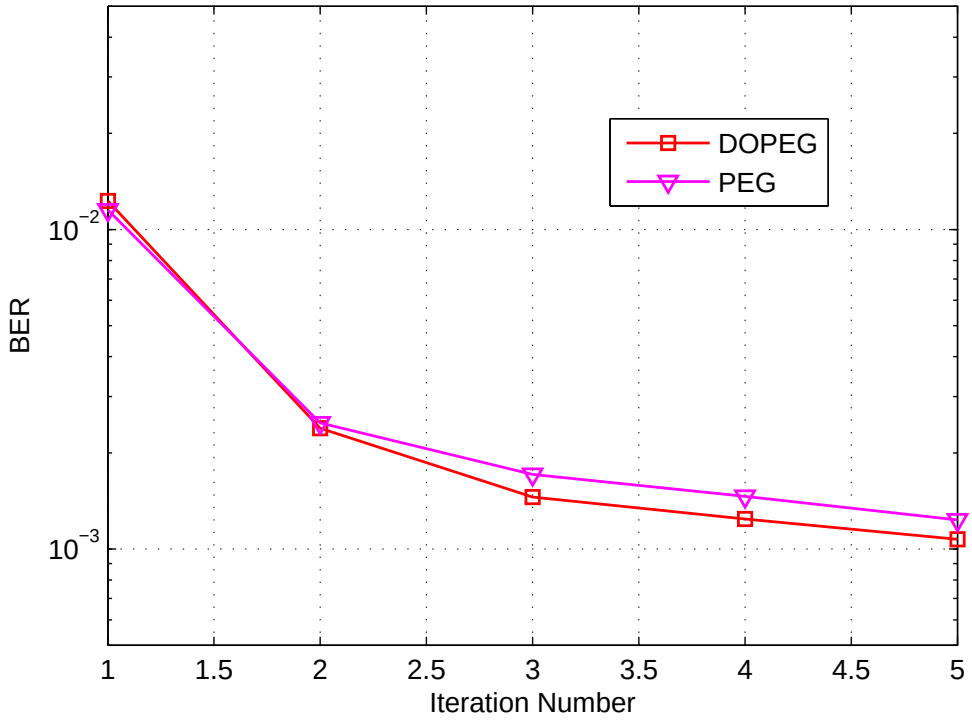


Figure 4.7: BER vs iteration number comparison of DOPEG and PEG generated codes for length $n = 248$ operating over an 8x8 MIMO system at Eb/N0 of -3dB

The DOPEG generated code which was tested here for comparison with the PEG constructed code was optimised over the range $[2 : 0.05 : 3]$ with 30 instances of message vectors transmitted. The decoder was operated to a maximum of 50 iterations in the optimisation stage of the DOPEG algorithm. This plot demonstrates that the benefits afforded by the DOPEG code when compared to the PEG code are consistent as the outer (detector) iteration number increases.

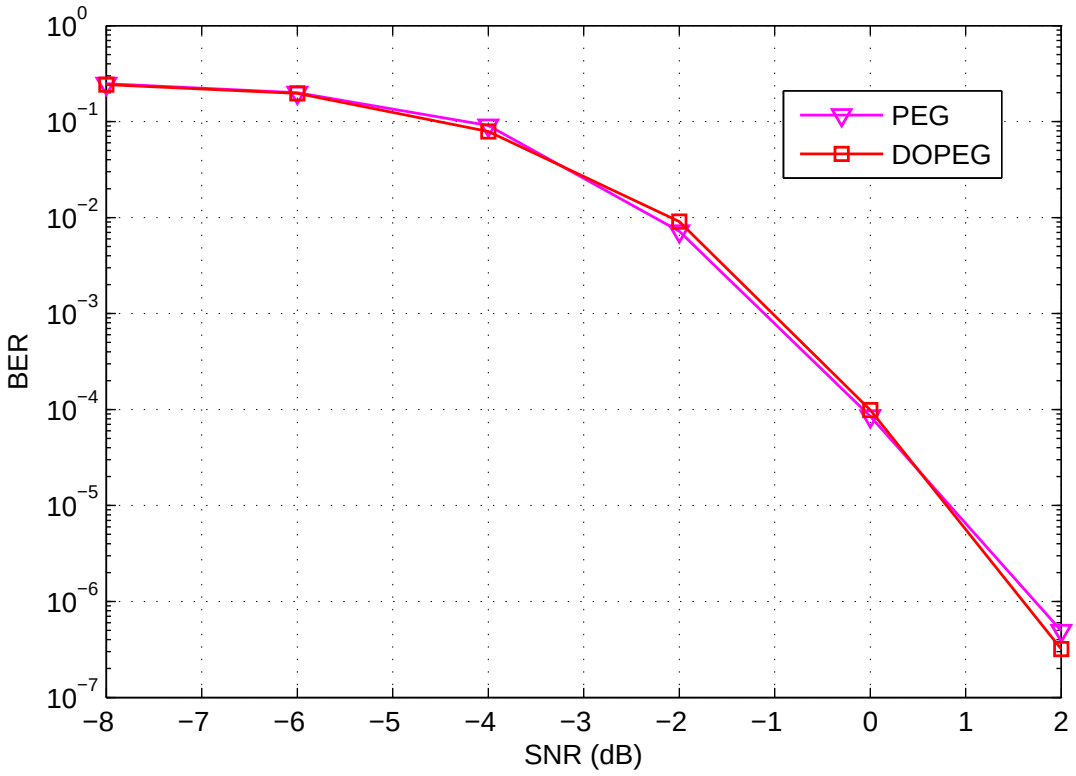


Figure 4.8: BER vs SNR length $n = 498$ DOPEG and PEG codes operating over an 6x6 MIMO system

Fig. 4.8 provides the BER curve for both PEG and DOPEG coded 8x8 iteratively detected and decoded MIMO systems. The outer (detector) loop was run for 5 iterations while the inner loop (SPA decoder) was run to a maximum of 10 iterations. As expected, some performance improvement is seen at the higher end of the SNR range examined. The gain is not as impressive as that seen in the sinple-input single-output (SISO) system with AWGN as seen in Fig. 3.3. It is possible that greater gain would be seen in the error floor region of the BER curve for this system.

The degradation in performance gain may be explained by the fact that the decoder optimisation of the code tested was carried out for the AWGN channel. Alteration of the DO operation for the flat Rayleigh fading channel model used may result in gains comparable to those seen for the SISO system and this provides a possible line of further investigation.

4.5 Chapter Conclusions

In this chapter, following the introduction of the system model for a spatially multiplexed MIMO system and a discussion of a number of different detection methods, an iteratively detected and decoded system with LDPC coding was described. This system was used to test the performance of the novel code presented in the Chapter 3. A number of other configurations of the system were also examined to provide a point of reference when examining the results. These included an uncoded spatially multiplexed MIMO system with a number of different sub-optimal detectors and the iterative (turbo) system on which the novel code is tested with PEG-LDPC codes of differing block lengths.

The simulation results, both BER against SNR and BER against iteration number plots, demonstrate that the novel codes of Chapter 3 provide performance improvements in the coded MIMO system under investigation.

Chapter 5

Conclusions

In this thesis a novel irregular LDPC code construction method was presented. This code was developed based on a number of key concepts in the area of LDPC codes and codes based on graphs in general. The goal in completing this thesis was to develop a clear understanding of the current state of the art in LDPC codes and to use this knowledge gained to design a novel LDPC code. The aim was to develop a construction method which can produce codes of practical block length with performance as close as possible to that of the ideal infinite block length code which exhibits a cycle-free decoding neighbourhood.

Among the best-performing codes currently available are those constructed by the PEG algorithm or its modifications. From an in-depth analysis of this algorithm, scope for a possible contribution was identified. The PEG edge placement algorithm effectively identifies the set of check nodes which will result in the creation of the cycle with greatest possible length when an edge is placed. The modifications [6][7] are based on finding a metric or metrics by which to compare the surviving candidates and then determining what order to apply these metrics in the pruning of the candidate check node set in order to find the survivor which will provide the best code performance.

A metric was developed which involved testing the operation of the code under the current graph setting with each candidate node in place and identifying the node which

provided the best performance. This metric was applied to the PEG algorithm and a significant improvement in performance was observed in short block length codes. At longer lengths, the rewards in terms of performance improvements, while still observed, are not quite as good for the parameters of the Decoder Optimisation (DO) procedure tested. However, as the gains observed are found in the error floor region of the BER curve, it is likely that further time intensive simulations would reveal greater gains at lower BERs. A possible source of further improvements over the base code construction method is to increase the number of instances of message and noise vectors generated for each candidate CN when performing the DO operation. This would increase the likelihood that the chosen CN is the one which provides the best overall performance. This also increases the complexity of the construction method.

The DO modification was then applied to the IPEG algorithm. The IPEG applies an extra pruning stage to the check node candidates of the original PEG in order to improve performance in the error floor area. The performance improvements seen from the DOIPEG over the IPEG were not as great as those observed when comparing DOPEG and PEG constructed codes. This is not unexpected as a candidate check node which is optimal according to the ACE metric is, naturally, more likely to produce better performance under SPA decoding and likewise the candidate with the highest DO metric is likely to have a high degree of connectivity to the rest of the graph. This large degree of graph connectivity explains why less of an improvement is noted in comparing DOIPEG and IPEG. In fact it can be seen that the DOPEG and IPEG algorithms produce codes with a similar level of improvement over those constructed by the PEG. This provided motivation to apply the DO operation to the IPEG extension to the algorithm. It is then seen that the resulting DOIPEG construction method provides improvement in performance over the IPEG algorithm. This is an encouraging result, as the IPEG construction method is among the best currently available for LDPC codes. Additionally, as was noted for the DOPEG case, a higher resolution of SNR points for Decoder Optimisation at a fixed SNR range is likely to produce greater performance improvements. Again, with extra effort at the design stage this modified algorithm can

produce worthwhile performance improvements at no extra cost of complexity during transmission and decoding.

Finally, we have considered the application of the proposed DOPEG designed code to MIMO systems. A spatially multiplexed MIMO system was overviewed and the iterative (turbo) detection scheme for joint detection and decoding was described. This system has been designed with minimum mean square error successive interference cancellation (MMSE-SIC) detection and LDPC channel coding and was tested for a number of different LDPC codes. The novel irregular DOPEG code was tested and some improvement in performance was noted.

Bibliography

[1] R. G. Gallager, “Low-density parity-check codes,” *IRE Trans. Inform. Theory*, vol. 20, pp. 21–28, Jan. 1962.

[2] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inf. Theory*, vol. 27, pp. 533–547, Sep. 1981.

[3] M. Luby, M. Mitzenmacher, A. Shokrohalli, D. Spielman, and V. Stemann, “Practical loss-resilient codes,” *Proc. 29th Annual ACM Symp. Theory of Computing*, pp. 150–159, 1997.

[4] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[5] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, “Regular and irregular progressive edge growth tanner graphs,” *IEEE Trans. Inf. Theory*, vol. 50, no. 1, pp. 21–28, Jan. 2005.

[6] H. Xiao and A. H. Banihashemi, “Improved progressive-edge-growth (PEG) construction of irregular LDPC codes,” *IEEE Commun. Lett.*, vol. 8, no. 12, pp. 715–717, Dec. 2004.

[7] D. Vukobratovic and V. Senk, “Generalized ACE constrained progressive edge-growth LDPC code design,” *IEEE Commun. Lett.*, vol. 12, no. 1, pp. 32–34, Jan. 2008.

[8] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. Commun.*, vol. 50, no. 8, pp. 1242–1247, Aug. 2004.

[9] G. Foschini and M. Gans, "On the limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, pp. 311–335, 1998.

[10] I. E. Telatar, "Capacity of multiantenna gaussian channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, Nov. 1999.

[11] D. MacKay, "Good error correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, pp. 399–431, Mar. 1999.

[12] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Construction of irregular LDPC codes with low error floors," *IEEE Conference on Comms., ICC*, vol. 5, pp. 3125–3129, May 2003.

[13] C. Di, , D. Proietti, E. Telatar, T. Richardson, and R. Urbanke, "Low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 48, pp. 1570–1579, Jun. 2002.

[14] T. Richardson, "Error floors of LDPC codes," *Allerton Conf. on Comm., Control and Comp.*, Oct. 2003.

[15] D. Vukobratovic, A. Djurendic, and V. Senk, "ACE spectrum of LDPC codes and generalized ACE design," *IEEE Conference on Comms., ICC*, pp. 665–670, Jun. 2007.

[16] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 638–656, Feb. 2001.

[17] M. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[18] S. Johnson and S. Weller, "A family of irregular LDPC codes with low encoding complexity," *IEEE Commun. Lett.*, vol. 7, no. 2, pp. 79 – 81, Feb. 2003.

[19] S. L. Y. Kou and K. Abdel-Ghaffar, "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, pp. 2711–2736, Nov. 2001.

[20] H. J. D. Divsalar and R. McEliece, "Coding theorems for turbo-like codes," *Proc. 36th Annual Allerton Conf. on Comm., Control and Computing*, pp. 201–210, Sep. 1998.

[21] A. K. H. Jin and R. McEliece, "Irregular repeat-accumulate codes," *Proc.2nd Int Symp. on Turbo Codes and Related Topics*, pp. 1–8, Sep. 2000.

[22] M. Yang, W. Ryan, and Y. Li, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 564–571, Apr. 2004.

[23] A. Abbasfar, D. Divsalar, and Y. Kung, "Accumulate-repeat-accumulate codes," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 692–702, Apr. 2007.

[24] D. Divsalar, S. Dolinar, C. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE Journal on Selected Areas in Comms.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.

[25] D. Divsalar, S. Dolinar, and C. Jones, "Reduced complexity mimo detectors for LDPC coded systems," *Proc. Int. Symp. on Inf. Th., ISIT*, pp. 1622–1626, 2005.

[26] M. Jankiraman, *Space-Time Codes and MIMO Systems*. Connecticut, USA: Artech House, 2004.

[27] S. Verdu, *Multiuser Detection*. Cambridge University Press, 1998.

[28] X. Wang and V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, Jul. 1999.

[29] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inf. Theory*, vol. 45, pp. 1639–1642, Jul. 1999.

[30] O. Damen, A. Chkeif, and J. C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Commun. Lett.*, vol. 4, no. 5, pp. 161 – 163, May 2000.

[31] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.

[32] G. Golden, G. Foschini, R. Valenzuela, and P. Wolniansky, "Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture," *IET Electronics Letters*, vol. 35, no. 1, pp. 14–16, Jan. 1999.

[33] R. Fa and R. C. de Lamare, "Multi-branch successive interference cancellation for MIMO spatial multiplexing systems: Design, analysis and adaptive implementation," *IET Communications*, 2010.

[34] N. Al-Dhahir and A. H. Sayed, "The finite-length multi-input multi-output MMSE-DFE," *IEEE Trans. Signal Process.*, vol. 48, no. 10, pp. 2921–2936, Oct. 2000.

[35] R. C. de Lamare and R. Sampaio-Neto, "Minimum mean-squared error iterative successive parallel arbitrated decision feedback detectors for DS-CDMA systems," *IEEE Trans. Commun.*, vol. 56, no. 5, pp. 778–789, May 2008.

[36] S. Moshav, "Multi-user detection for DS-CDMA communications," *IEEE Commun. Mag.*, vol. 34, pp. 124–136, 1996.

[37] G. Woodward, R. Ratasuk, M. Honig, and P. Rapajic, “Minimum mean-squared error multiuser decision-feedback detectors for DS-CDMA,” *IEEE Trans. Commun.*, vol. 50, no. 12, Dec. 2002.

[38] D. Wuebben, R. Boehnke, V. Kuehn, and K. D. Kammeyer, “Near maximum-likelihood detection of mimo systems using MMSE-based lattice reduction,” *Proc. IEEE Int. Conf. Commun., (ICC04)*, Jun. 2004.

[39] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding: Turbo-codes,” *IEEE Conference on Comms., ICC*, pp. 1064 – 1070, May 1993.

[40] A. Matache, C. Jones, and R. D. Wesel, “Reduced complexity MIMO detectors for LDPC coded systems,” *IEEE Military Communications Conference*, vol. 2, pp. 1073–1079, Oct. 2004.

[41] T. Abe and T. Matsumoto, “Space-time turbo equilization in frequency-selective mimo channels,” *IEEE Trans. Veh. Technol.*, vol. 52, no. 3, pp. 469–475, May 2003.

[42] D. Reynolds and X. Wang, “Low-complexity turbo-equalization for diversity channels,” in *Signal Processing, Elsevier Science Publishers*, 2000, pp. 989–995.